ON OPTIMIZING ASSEMBLY PLANS FOR A ROBOTIZED PRINTED CIRCUIT BOARD ASSEMBLY CENTER

by

CHI-MING CHANG, B.S., M.S. in I.E.

A DISSERTATION

IN

INDUSTRIAL ENGINEERING

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

DOCTOR OF PHILOSOPHY

Approved

December, 1986

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor, Dr. Milton L. Smith, for his continuous guidance, supervision and support that led to the completion of this research.

My gratitude also goes to other members of my committee--Dr. William M. Marcy, Dr. Eric L. Blair, Dr. Ralph R. Bravoco, and Dr. Brian K. Lambert--for their suggestions and comments.

Texas Instruments personnel at Austin, Texas were gracious in giving their time in discussing the problem and in providing data. Mr. Neal Klubak's assistance is especially appreciated.

I appreciate the encouragement and help of my dear friends Mr. Peerayuth Charnsethikul and Mr. Yi-Fong Wung. Mr. Charnsethikul's detail introduction to Traveling Salesman Problem provided a good Foundation for the development of the mathmatical models in this research. Mr. Wung kindly help me to provide the final version of the dissertation after I left Texas Tech.

I would like to thank my parents and my wife Chyong-Yau, for their everlasting moral support and encouragement.

This research was supported in part by Productivity in High Tech Manufacturing research project within the Texas Advanced Technology Research Program.

ABSTRACT

The purpose of this research was threefold. First, a set of parameters was defined for use in analyzing the assembly planning problem in robotized printed circuit board (PCB) assembly centers. Second, a number of mathematical models were developed to optimize assembly plans for a robotized PCB assembly center. Third, an integrated heuristic method was developed to obtain near-optimal assembly plans.

The robotized PCB assembly center of interest consists of a moving X-Y table, a moving feeder carrier and a traveling pick-and-place robot. The assembly planning problem is one of determining the assembly sequence and assignment of feeders to optimize two prioritized objectives: minimizing assembly cycle time and minimizing total X-Y table travel time.

Three groups of parameters were defined through analysis of the problem. These are: (1) non-timing parameters, (2) timing parameters and (3) priority parameter. Based on the parameters, 10 different assembly environment classes were identified.

A number of mathematical models were developed. Among them, four models were of traveling salesman problem nature

but with modification on the associated cost matrices. The remaining models were more complicated and were non-linear. These models were linearized to obtain 3 modules.

An integrated heuristic method of five stages was developed to quickly reach a near-optimal assembly plan. A set of experiments was performed to evaluate the performance of this integrated heuristic method.

When the number of feeders is no less than the number of components total delay can always be zero while total X-Y table travel is less than 1% above the best known solutions. When the number of components is greater than the number of feeders and total X-Y table travel time receives higher priority, the resultant assembly plan can have both performance measurers less than 1% above the best known solutions. For the remaining classes, the resultant assembly plans always have zero delay while total X-Y table travel time is no more than 3% above the best known solutions if Stage III and Stage V are used.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

1. AC: Accuracy capability of the X-Y table.
2. BD(i,j): The board delay for assembly step (i,j).
3. BDI(i,j): The board delay of assembly step (i,j) under the influence of pick delay occurring in the assembly sequence.
4. BDM: Board delay matrix.
5. BE: Board extreme moving time.
6. CHP1: Nearest neighbor method.
7. CHP2: Farthest insertion method.
8. CHP3: Nearest neighbor method first, then 2-opt method.
9. CHP4: Farthest insertion first, then 2-opt method.
10. CHP5: Nearest neighbor method first, then 3-opt method.
11. CHP6: Farthest insertion method first, then 3-opt method.
12. CHP7: Nearest neighbor method first, 2-opt method second, then 3-opt method.
13. CHP8: Farthest insertion method first, 2-opt method second, then 3-opt method.
14. CM: Cost matrix.
15. D: Percentage of assembly cycle time above best known solution.
16. DC: Delivery capability of the feeder carrier.
17. FAS: The set of feasible assembly steps.
18. FBDM: Feasible board delay matrix.

19. FPDM: The feasible pick delay matrix.

20. FREQ(I,J): The number of assembly steps from component type I to component J plus the number of assembly steps from component type J to component type I.

21. FREQM: Interaction frequency matrix.

22. FXYM: Feasible X-Y table travel time matrix.

23. HPO: Priority parameter.

24. HPO=A: The assembly planning problem is of form A. It is desired to minimize total assembly cycle time first and then to minimize total X-Y table travel time.

25. HPO=B: The assembly planning problem is of form B. It is desired to minimize total X-Y table travel time first and then minimize total assembly cycle time.

26. IFAS: The set of infeasible assembly steps.

27. I,J: The index number for component types. I=1,2,....,NTT. J=1,2,.....,NTT.

28. IM: Inversion cost matrix.

29. M: The travel time between two extreme feeders in the feeder carrier.

30. MAC: The minimal assembly cycle time without any delay.

31. N: Number of components in PCB to be assembled by the PCB assembly center.

32. NF: Number of available feeders in the feeder carrier.

33. NTT: Number of component types in PCB to be assembled by the PCB assembly center.

34. OB(i,j): The X-Y table travel time required for assembly step (i,j).

35. P(I): The position number of the feeder which is assigned the component type I.

36. P(I$): The position number of the feeder which contains component type I determined from the current local optimal assignment.

37. P(i): The position number of the feeder to which the ith component is assigned.

38. PD(I,J): The pick delay time for an assembly step (I,J).

39. PD(i,j): Pick delay of assembly step (i,j).

40. PDCM: 2-switching total pick delay matrix.

41. PDI(i,j): The pick delay of assembly step (i,j) under the influence of board delay occurring in the assembly sequence.

42. PSUM(I): Partial interaction sum of an unassigned component type.

43. PT: The time needed for robot to pick/place a single component.

44. R: The use of the stage depends on the result of Stage II.

45. RINF: Extremely large value.

46. RT: Robot normal round trip time.

47. SF1: The scaling factor which is the summation of the N largest X-Y table travel times of the feasible assembly steps.

48. SF2: The scaling factor which is the summation of the N largest board delays of the feasible assembly steps.

49. SPT: Shortest placing related timing parameter.

50. SPT=AC: Accuracy capability (AC) is the smallest among the three placing related timing parameters.

51. SPT=BE: Board extreme moving time (BE) is the smallest among the three placing related timing parameters.

52. SPT=RT: Robot normal round trip time (RT) is the smallest among the three placing related timing parameters.

53. TBD: The minimal board delay obtained at Phase I.

54. T(NTT+C): The component type to which the (NTT+C)th component belongs.

55. TL: Percentage of total X-Y table traveling above best known solution.

56. TOB: The total X-Y table travel time obtained in Phase I.

57. TPD: The total pick delay obtained in Phase I.

58. TSUM(I): Total interaction sum of a component type.

59. U: The use of the stage is determined by user.

60. UL: The total time of unloading and loading.

61. UT: Unit delivery time of the feeder carrier.

62. V: The speed of X-Y table.

63. XYM(i,j): X-Y table travel time matrix.

64. (HPO=A/N>NF/SPT=AC): An assembly environment class in which
    (1) The assembly planning problem is of form A. It is desired to minimize total assembly cycle time first and then to minimize total X-Y table travel time.
    (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier.
    (3) Accuracy capability (AC) is the smallest among the three placing related timing parameters.

65. (HPO=A/N>NF/SPT=BE): An assembly environment class in which
    (1) The assembly planning problem is of form A. It is desired to minimize total assembly cycle time first and then to minimize total X-Y table travel time.
    (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier.
    (3) Board extreme moving time (BE) is the smallest among the three placing related timing parameters.

66. (HPO=A/N≤NF/SPT=RT): An assembly environment class in which
    (1) The assembly planning problem is of form A. It is desired to minimize total assembly cycle time first and then to minimize total X-Y table travel time.
    (2) The number of components in the PCB is less than or equal to the number of feeders in the feeder carrier.
    (3) Robot normal round trip time (RT) is the smallest among the three placing related timing parameters.

67. (HPO=A/N>NF/SPT=RT): An assembly environment class in which
    (1) The assembly planning problem is of form A. It is desired to minimize total assembly cycle time first and then to minimize total X-Y table travel time.
    (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier.
    (3) Robot normal round trip time (RT) is the smallest among the three placing related timing parameters.

68. (HPO=AB/N≤NF/SPT=AC): An assembly environment class in which
    (1) The priority order of the two objectives can be either to minimize total assembly cycle time first and then to minimixe total X-Y table travel time or to minimize total X-Y table travel time first and then to minimize total cycle time.
    (2) The number of components in the PCB is less than or equal to the number of feeders in the feeder carrier.
    (3) Accuracy capability (AC) is the smallest among the three placing related timing parameters.

69. (HPO=AB/N≤NF/SPT=BE): An assembly environment class in which
    (1) The priority order of the two objectives can be either to minimize total assembly cycle time first and then to minimixe total X-Y table travel time or to minimize total X-Y table travel time first and then to minimize total cycle time.
    (2) The number of components in the PCB is less than or equal to the number of feeders in the feeder carrier.
    (3) Board extreme moving time (BE) is the smallest among the three placing related timing parameters.

70. (HPO=B/N≤NF/SPT=RT): An assembly environment class in which
    (1) The assembly planning problem is of form B. It is desired to minimize total X-Y table travel time first and then to minimize total assembly cycle time.
    (2) The number of components in the PCB is less than or equal to the number of feeders in the feeder carrier.
    (3) Robot normal round trip time (RT) is the smallest among the three placing related timing parameters.

71. (HPO=B/N>NF/SPT=AC): An assembly environment class in which
    (1) The assembly planning problem is of form B. It is desired to minimize total X-Y table travel time first and then to minimize total assembly cycle time.
    (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier.
    (3) Accuracy capability (AC) is the smallest among the three placing related timing parameters.

72. (HPO=B/N>NF/SPT=BE): An assembly environment class in which
    (1) The assembly planning problem is of form B. It is desired to minimize total X-Y table travel time first and then to minimize total assembly cycle time.
    (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier.
    (3) Board extreme moving time (BE) is the smallest among the three placing related timing parameters.

73. (HPO=B/N>NF/SPT=RT): An assembly environment class in which
    (1) The assembly planning problem is of form B. It is desired to minimize total X-Y table travel time first and then to minimize total assembly cycle time.
    (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier.
    (3) Robot normal round trip time (RT) is the smallest among the three placing related timing parameters.

74. (I,J): The assembly steps from component type I to component type J or from component type J to component type I.

75. (i,j): The assembly step with componnent i immediately proceeding component j.

CHAPTER I

STATEMENT OF THIS RESEARCH

Introduction

A large number of industrial and commercial products contain at least some electronic components. Non-electronic products like motors, typewriters, and clocks have gained new functionality and consumer interest as the result of electronic components that facilitate the control of these devices.

All electronic devices contain various types and configurations of electronic components. The most common method employed for fixing the components in electronic equipment is through the use of printed circuit boards (PCBs).

A printed circuit board usually serves three quite distinct functions: (1) it provides the necessary mechanical support for the components in the circuit; (2) it provides the necessary electrical interconnections and (3) it frequently bears some form of legend which identifies all components it carries (55).

Although the inserted circuit board is often a subassembly of a larger and considerably more complicated

electronic device, the direct costs associated with printed circuit boards represent a major portion of the total variable cost.

On printed circuit board lines competition, accelerated new product introduction, increasing board densities, stiffer quality requirements, and price/cost pressures are forcing manufacturing managers to rethink their entire manufacturing strategy. The new strategy is a system approach to automate the entire production cycle including PCB design, bare PCB manufacturing, PCB assembly, and PCB testing (39).

In PCB assembly, the new strategy means being able to call up a new program and start up a whole new board assembly sequence in seconds to minimize the assembly cycle time for the PCB. The general method of automating the PCB assembly processes is a combination of two mutually complementary approaches: the development and application of assembly robots and the development of new design/manaufacturing technology of PCB components for ease of robotic assembly.

One major redesign approach in PCB assembly has been to decrease component size and difficulty of assembly tasks while increasing the power and versatility of the components. The major breakthroughs in this approach include:

1. The use of dual in-line packages (DIPs) where many functions may be included in one chip thus reducing assembly requirements to the insertion of one component rather than several components, each with lesser functions.
2. Standardization of DIPs in two forms. The first form is in electronic functioning, meaning that the 'same' integrated circuit (IC) is available from several suppliers. The second form of standardization is DIP format. The standardized format simplifies the insertion process as parts variety has been decreased.
3. Development of surface mount devices (SMDs) to replace traditional DIPs in through-hole technology. Surface mount technology offers several advantages. These include significant size and weight reduction, a reasonable solution for high pin count ICs, improvement in electrical performance, and an easier approach for automating the assembly tasks since through-hole DIPs insertion is replaced by placing SMDs right on prespecified positions.

Robots have attracted attention of many researchers for the past two decades. A substantial amount of recent research has been directed toward development of industrial robots in the fields of electro-mechanical capabilities,

sensing devices, gripper design and computer controls; relatively little research investigating the operational problems associated with applications of this technology has been published.

In a robotized PCB assembly center, electronic components are selectively picked by one or more robots and are inserted or attached to their respective positions in the PCB. Assembly tasks involve repeated sequences of pick-move-insert (place) operations.

Assembly cycle time, PCB quality and the maintainance cost of the robotized PCB assembly center may be directly or indirectly affected by the planning and control of the assembly sequences and the locations of component feeders. Thus, assembly planning for robotized assembly center is an important operational problem. This problem has added value when the cost pressures that are faced by companies in the United States are considered. Also, the large number of PCBs assembled annually make this an important problem. An assembly planning problem for a robotized PCB assembly center was analyzed and solved in this research.

Purpose of This Research

Robotized printed circuit board assembly centers are becoming increasingly important to the electronic industry as a method of improving productivity. Various techniques are currently used to find a feasible assembly plan in robotized printed circuit board assembly centers. An application engineer either uses a trial and error method to find a feasible assembly operation sequence without paying attention to the coordination of the robots and other constituents of the assembly center (e.g. component feeders) or arranges the assembly plan according to the instructions of the system computer. The performance of a robotized PCB assembly center may largely depend on the planning and control of the assembly sequences and the assignment of components to the feeders.

The purpose of this research was threefold. First, a set of parameters was defined for use in analyzing assembly planning problems in robotized PCB assembly centers. Second, a number of mathematical models were developed to optimize assembly plans for a robotized PCB assembly center by considering coordination of robot movements, X-Y table traveling and the movements of feeder carrier. Third, an integrated heuristic method was developed to obtain near-optimal assembly plans of the robotized PCB assembly center.

The Robotized PCB Assembly Center

The robotized PCB assembly center analyzed in this research consists of the following building blocks:

1. A pick-and-place mechanism with one robot mounted on an overhead rail-track. The robot can move between the fixed pick point and the fixed place point. At the fixed pick point, the robot arm can move vertically in the Z direction to pick up required components. At the fixed place point, the robot arm can move vertically in the Z-direction to place components on the PCB.
2. A movable X-Y table on which the PCB is fixtured. The X-Y table can move linearly in the X-Y plane.
3. A straight-line feeder carrier which can move linearly along the straight line such that the required components are delivered to the pick point. The feeder carrier, located at one side of the X-Y table, contains a number of equally-spaced feeders. Each feeder may contain a number of components of the same type.

Assembly operations conform to the following assumptions:

1. All robot travel and manipulation times are known.

2. PCB unloading/loading is done by a mechanism outside the assembly center; during this period of time all the other operations are temporarily halted.
3. No breakdowns occur.
4. The system operates deterministically .

The layout of this robotized PCB assembly center is presented in Figure 1.

There may be many different types of PCBs to be processed by the assembly center. PCB types may be different either in board size, in component layout, or number, size and function of components. The assembly cycle starts with the loading of a PCB on the concave portion of the movable X-Y table. Once the loading operation is finished, the robot moves to the pick point while the feeder carrier delivers the first required component feeder to the pick point, picks up one component and moves to the place point while the PCB table is moving the PCB so that the location to receive the first component is under the place point. When the robot and the board have moved to the correct point, the robot places the component on the PCB. The robot then moves back to pick point and picks up the second component prespecified in the assembly sequence immediately if the feeder containing that component has been moved to the pick point. Otherwise, the robot must wait for

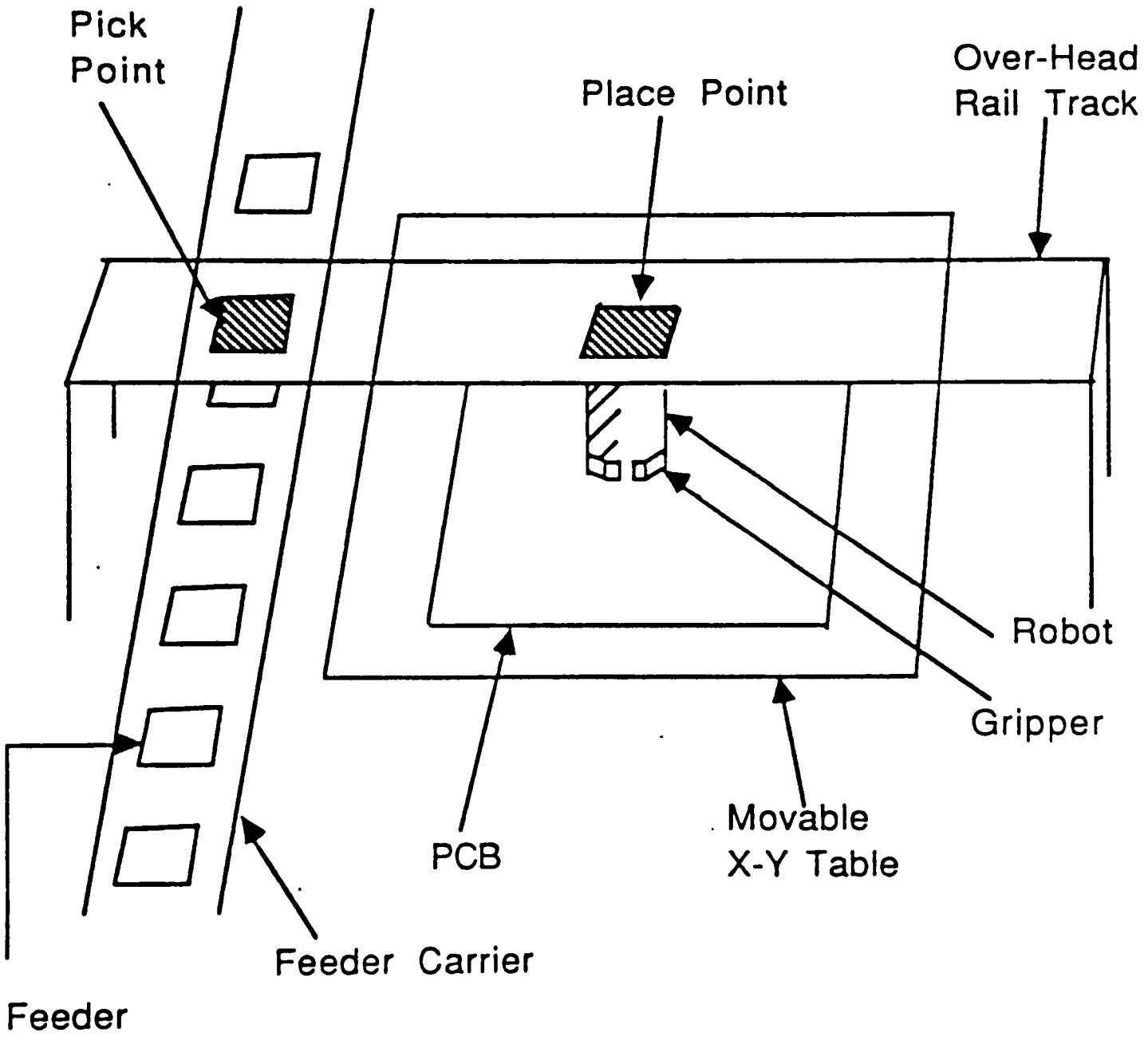


Figure 1: Layout of The Robotized PCB Assembly Center

this component. Operations continue as described until all the components are placed in the PCB. Then this PCB is unloaded and another PCB is loaded.

This robotized PCB assembly center is a simplified version of a two-robot PCB assembly center that is used by many companies performing electronic assembly. The two-robot PCB assembly center differs from the one-robot PCB assembly center considered in this research in that there are two synchronized robots in the two-robot PCB assembly center instead of one robot, and there are two straight-line taped part feeder carriers, one on each side of the X-Y table, instead of one feeder carrier.

The assembly planning problem associated with a two-robot PCB assembly center is more complicated than that of the one-robot PCB assembly center. However, it is believed that the analysis of one-robot PCB assembly center can provide theoretical basis for further research in two-robot PCB assembly center. As a pioneer study in this field, this research will focus on the one-robot PCB assembly center to initiate a structure for analyzing the assembly planning problem in robotized PCB assembly centers.

The Assembly Planning Problem

The assembly planning problem in a robotized assembly center concerns the determination of the assignment of components to the feeders of the taped part feeder carrier and the determination of the assembly sequence to optimize two objectives. The first objective is to assemble PCBs, under required placement accuracy, in the minimum cycle time. On the other hand, it is believed that the more distance the X-Y table travels the less positioning accuracy capability it posses due to the wear accumulated. The second objective is to minimize the PCB X-Y table movements and thereby to reduce wear on the PCB X-Y table movement mechanisms.

The priority of the two objectives depends on managerial decision. For example, if assembly capacity is sufficient to handle the volume, then management might be willing to give maintainance higher priority. That is, the manager would like to minimize the total X-Y table travel time first and then to minimize assembly cycle time. On the other hand, if assembly capability is fully utilized, then minimizing assembly cycle time may receive higher priority. That is, the manager might be willing to minimize the assembly cycle time first and then to minimize total X-Y table travel time.

Therefore the assembly planning problem may assume one of two forms according to the priority order of the two objectives. These two forms are:

Assembly planning problem form A: The problem of determining the assignment of components to the feeders in the feeder carrier and the assembly sequence to minimize assembly cycle time first and then to minimize total X-Y table travel time.

Assembly planning problem form B: The problem of determining the assignment of components to feeders in the feeder carrier and the assembly sequence to minimize total X-Y table travel time and then to minimize assembly cycle time.

Outline of This Research

The following outline summarizes this research. In Chapter 2, previous research on managerial problems in robot applications is reviewed. Since this research is related to the Traveling Salesman Problem (TSP), previous research on TSP is reviewed. In Chapter 3, the assembly planning problem is analyzed. A set of system parameters is derived and is used to classify different assembly environment classes. The performance characteristics of each assembly environment class are identified.

Mathematical models for obtaining optimal assembly plans under different conditions are developed in Chapter 4. An integrated heuristic method is developed to obtain near-optimal solution for the assembly planning problem. Descriptions of this integrated heuristic method are presented in Chapter 5. Performance of this integrated heuristic method is evaluated and discussed in Chapter 6 along with an example to illustrate the application of the integrated heuristic method.

The assembly planning problem and the solution procedure developed in this study are summarized, and recommendations for future research are suggested in Chapter 7.

CHAPTER II

LITERATURE REVIEW

Industrial Robots

The Robot Institute of America (RIA) defines a robot as a "reprogrammable, multifunctional, manipulator designed to move materials, parts, tools or special devices through variable programmed motions for the performance of a variety of tasks (53)". Reprogrammability makes it possible to adapt the robot in dynamic manufacturing environments where the switch from one part or batch to another can be effected by changing the program of instructions. Multifuctionality enables the robot to perform a wide variety of tasks under the program instructions. With manipulator capabilities the robot is differentiated from other forms of programmable automation systems, such as numerical control machines.

Robots can be seen as mechanical copies of humans which combine intelligence, manipulation ability and the end effector (12). The end effector may be a spray painter, a tool or a gripper. Depending on the intelligence that is provided within the computer controller and its computer, a robot can be highly smart and sophisticated.

Industrial robots are used in an incredible variety of applications. There are few processes where robot can not be used. Robots appear in space, in the sea, in coal mines, inside nuclear reactors, and in manufacturing (42).

The major application areas of industrial robots which have been reported include (51):

1. Pick-and-place type of operations, in which case point-to-point controlled devices are usually sufficient, such as machine loading and unloading, simple palletizing, glass handling, press work, heat treating, coating and electroplating, investment casting, die casting, forging, and metal coating.
2. Welding robots, which are the largest in terms of population since the automotive industry employs the greatest number (over 40% of the total robot population). Both point-to-point controlled and continuous-path controlled robots are widely applied for spot welding, arc welding, seam welding, flame cutting, laser welding, and plasma cutting.
3. Machining robots, which require a rigid body and arm structure with adequate positioning repeatability. The machining robots often have to change tools and/or hands to accommodate different jobs. Machining operations include: drilling, routing,

sheet metal fabrication, and composite materials manufacturing.

4. Cleaning and deburring applications with special purpose tools. Such robot applications often need adaptive force feedback sensing as well as automated robot tool changing.
5. Spray painting.
6. Automated assembly and inspection, which is probably the most interesting and fastest growing area of application. This is due to the potential of the growing market of the assembly industry and its complexity in programming, tooling, sensory feedback processing, interfacing with other devices and communicating with remote computers.

Kondoleon (34) examined assembly tasks in terms of direction of work activity and found that a single vertical motion accounted for the majority of tasks; in other words, assembly consists of single linear motion that move components vertically downwards from a rest position above an item, so that the two objects are in physical contact. Other motions associated with assembly tasks are the pick and place activities that transport the individual items from presentation devices to the preassembly position. In general, non-linear and non-vertical motions are few, and

their use within assembly tasks is more a function of the geometry of the robot rather than the requirement of the assembly tasks.

Holmqvist (27) reported that there are three main robot configurations used in assembly applications. These are: (1) rectangular coordinate types, (2) horizontally jointed types (SCARA), and (3) universal types (revolute). He also introduced a pendulum robot concept for assembly of small parts in medium production volumes.

Owen (46) indicated that a number of non-servoed pick and place units are fully capable of performing many assembly tasks. These units are usually in one of the three forms (cartesian, gantry or cylindrical) and can be supplied as modular parts that can be configured into a desired arrangement. These pick and place units are also referred to as pick-and-place robots.

Owen also cited that the nonservoed pick and place devices have three distinct advantages over servoed robots. They are:

1. Repeatability is much better than a 'true' robot, typically +/- .02 mm versus +/- .05 mm for the majority of commercially available assembly robots.
2. They are faster in that they are penumatically operated and 'decelerated' by smashing against the

end stops, which are cushioned to absorb the energy of impack so no damage is done to the system.

3. Cost is between one-tenth and one-half of that of a servoed assembly robot.

Assembly operations are generally conceded to offer the greatest potential for use of industrial robots. Two broad category of robotic assembly include electronic assembly and mechanical assembly. According to a Delphi Survey of Robotics and Factory Automation conducted by the University of Michigan for the Society of Manufacturing Engineers (57), electronic assembly robots will account for 12% of the total robot sales in 1990 and 16% in 1995. This implies that robots will play an important role in future electronic assembly industry.

Studies in Managerial Problems of Robot Applications

While a substantial amount of recent research has been directed toward updating existing technology as well as developing new technology to increase and/or enrich capabilities of industrial robots, relatively little research has investigated managerial problems associated with application of this technology, perhaps because applications are diversed and share few common aspects that

can be easily recognized and structured for analysis. Generally speaking, research in managerial problems can be classified into two categories: (1) robot selection and (2) operations planning.

Robot Selection

Robot selection deals how to select a robot i, with attributes (X1, X2,---, Xn) that can perform a specified task, with characteristics (X1, X2,---, Xm), at the minimum production cost.

Several research papers have contributed in determining the appropriate robot for a given task. Paul and Nof (49) introduced a Robot Time and Motion methodology (RTM) for the analysis and design of robot work. Compared to the Methods Time and Measurement (MTM) for humans, the authors found that while the MTM methodology may be desired for work method analysis for robots, different motion time elements and parameters should be included for robot work. The result of their research, RTM, is a high level task description language which is formulated to evaluate robot ability to perform a given task, estimate task execution time, and compare alternate robot work methods. This research paved the way for later research in robot selection problems.

Graves and Whitney (21) proposed a mathematical programming procedure for equipment selection and system evaluation in programmable assembly. Four years later, Graves and Lamar (20) made some extension of the station selection and task assignment problem discussed by Graves and Whitney.

Graves and Lamar formulated this problem as a zero-one integer program and described a procedure for seeking lower and upper bounds to the optimal value of the integer program. This integer program can be applied to situations where the precedence relationship of the subassemblies are known.

Nof, et al. (43) carried the Paul and Nof study further. They reported on a comparative robot-man skills and characteristics guide for choosing between robot and man. Once the decision was made to use robots, the authors then provided information on the characteristics of the robot that should perform the task. Although the study is general in nature, it does provide a guideline for evaluating alternative robot models for a given tasks. Kamali, et al. (32) proposed a framework for task assignment among humans, robots, and automation.

Nof and Lechtman (44) described the Paul and Nof RTM system in detail. The three main components of the model are:

1. RTM,
2. Robot performance models, and
3. The RTM analyzer.

The methodology has the capability to:

1. Systematically specify a work method for a given robot in a simple, straight forward manner.
2. Apply computer aids to evaluate a specified method by time-to-perform, number of steps, positioning tolerance and other requirements so that alternatives work methods can be compared.
3. Repeat robot evaluation for alternative robot models.

Recently Offodile (45) developed a user oriented computer-aided robot selection procedure (CARSP) to help solve the robot selection problem. The major steps of Offodile's approach include:

1. Choosing appropriate robot characteristics for the creation of Robot Descriptive Database (RDD).
2. Developing a Robot Coding and Classification System (ROBOCODE) by using the robot characteristics.
3. Translating task variables to those of the robots in the RDD.
4. Developing a procedure for matching task variables to those of the robots in the RDD.

5. Choosing the right robot through an economic analysis.

Operations Planning

This problem is to determine the best production plans in the following areas:

1. layout of robotized production systems,
2. operations sequence of robotized production systems.

Systematized methodologies for designing robotic cells have been described by Chang and Goldman (5), Fisher et al. (17), and Young et al. (62). Sarin and Wilhelm (54) proposed prototype models for two-dimensional layout design of robotic systems.

Considerations important in applying robots to tend machines are discussed by Bjork (4), Holmes (28 and 29) and Warnecke, et al. (60). Simulation has been applied by Heginbotham, et al. (24) and Warnecke, et at. (59) to study task sequencing issues. Baumann, et al. (1) derived models which define robot and machine utilization for tending applications in which machines are served sequentially. Wilhelm and Sarin (61) initiated a structure for analyzing problems involved in sequencing the operations that must be performed by a robot to tend machines in a cell. They concluded that a change in number of machines in the cell

would not change the basic structure of the problem. In contrast, a change in number of part types would change the fundamental structure of the problem and would dramatically increase problem difficulty. Optimal sequencing procedures are described for several fundamental cases. Models were developed to prescribe solutions in more advanced cases.

In the field of PCB assembly, axial lead components, integrated circuits in dual-in-line packages, transistors, connector pins, and jumper wires can be inserted by many different methods. Titlebaum (58) classified these methods into two categories: manual insertion and automatic insertion. Two types of manual aids are used in manual insertion. They are optical aids and insertion aids. An automatic insertion machine inserts components and clinches and/or cuts the lead under the board. Four types of automatic insertion machines were citied by Titlebaum. These are bench inserters, pantograph inserters, computer controlled insertion machines, and conveyor systems.

Automatic insertion of electronic components is an efficient method of assembling printed circuit boards. The process involves programming an automatic insertion machine to take components from a reel and place them in the PCB in the proper locations. However, the components must be in the order expected by the automatic insertion program, a task accomplished with an automatic sequencer.

Since hundreds of different types of components are used in PCBs, this necessitates changes in the sequencer component and tooling setups from board to board. These setup changes increase the cost of producing PCBs automatically and reduce the amount of available sequencer production time. Richardson (52) investigated the sequencer setup problem and formulated the problem as a traveling salesman problem in which each tooling setup is analogous to a point in space. The number of component changes required to go from one setup to another is the distance between the points. The problem was solved by "closest unvisited city" heuristic rule.

Randhawa, et al. (50) indicated that the sequencing costs are affected by two sequential decisions: sequencer mix or the size and the number of sequencers required to meet the production requirements, and the processing schedule at each sequencer. An integer programming model was developed to determine the optimum sequencer mix to minimize the sequencing costs while satisfying the production requirements. The integer programming model was then solved by means of the branch and bound algorithm on a multi-purpose optimiztion system (MPOS).

Parallel to the development of surface mount devices is the development of the robotized PCB assembly centers which

may actually eliminate the need for a sequencer. Six robotized PCB assembly centers have been developed and are competing in the growing market (63).

Drezner and Nof (13) investigated pin picking and insertion plans for assembly robots. This is the first published research which deals with the operation planning problem in assembly application of industrial robots. They define the assembly planning problem as the problem of how to physically arrange assembly cell components, particulally those dependent upon the products which are being assembled (e.g. bin cell locations), with the objective to optimize pick, place, and insert motions. The basic problem, called the Simple Assembly Plan problem (SAP), is formulated as a traveling salesman problem for the situation where the number of bin cells is equal to the number of parts in the assembly place. Extensions to this problem are heuristically solved by seperating the SAP problem into Bin Assignment Problem (BAP) and the Pick-Insert Sequencing problem (PIS). That is , they obtain the best linear assignment of the BAP, and solve the best sequence of the PIS under the bin cell assignment obtained previously.

Studies in Traveling Salesman Problem

The traveling saleman problem may be stated as follows: Given a set of n cities and a cost matrix (representing distances, money costs, travel times, etc.,) between the cities, it is required to determine the least-cost tour passing once and only once through each city. Further, if the cost matrix is symmetrical, the problem is called a symmetrical traveling saleman problem; otherwise it is called an asymmetrical traveling saleman problem. Throughout this chapter, only the symmetrical traveling saleman problem is treated because the assembly planning problem investigated in this research deals with symmetrical cost matrices.

The traveling saleman problem is perhaps the most celebrated of all discrete optimization problems. The task of developing solution procedure to this problem has attracted substantial interest from mathematicians, operations researchers, and computer scientists for a long time (48).

Much of the problem's attraction stems from the mathematical curiosity it creates while at the other end of the spectrum, interest is somewhat more pragmatic. Here the need to solve large scale traveling saleman problems arises

in many areas: vehicle routing, computer wiring, and job sequencing (35).

Research associated with traveling saleman problem can generally be classified into 2 directions. The first direction is from theoretical point of view. Karp (33) has shown the traveling saleman problem belongs to the class of NP-Complete or "hard" combinational optimization problems which to date can not be solved by polynomially bounded algrithms. However, theoretical researchers in traveling saleman problem insist on designing "good" solution methods to obtain optimality of the solution. The second is from a practical point of view in which researchers insist on fast heuristic solution methods and accept the possibility of a suboptimal solution.

Optimal solution methods for traveling saleman problems were founded either on the basis of integer programming or on the basis of dynamic programming. The recursive technique of dynamic programming has been applied to the TSP (2 and 25). Due to its enormous storage requirements, dynamic programming can solve only relatively small problems.

Integer Programming Approach

The first integer programming approach towards a solution of the TSP was proposed by Dantzig, et al. (11). Their idea was to generate a "good" solution heuristically, and to formulate the TSP as a linear programming problem in zero-one variables, and then to try to prove optimality of the heuristically obtained tour using cutting planes. Their procedure contained interactive parts and did not result in a straight forward algorithm.

A number of branch and bound algorithms (14, 26, 32, 38 and 56) have been developed for this class of problem. In the Little, et al. algorithm (38), at every stage a variable is chosen and fixed at its two alternative values thus seperating the current subset of feasible solution into two other subsets. The assignment problem which is used to evaluate a lower bound for each of the two subsets is not actually solved, but the lower bound to the assignment problem is obtained (which is also a lower bound to each of the two subsets). The seperation process continues until either a feasible solution is discovered or the current subset is fathomed. Loop constraints are treated implicitly by fixing the variables at 0 after every seperation stage that might create subloops.

In Eastman's algorithm (14) , the assignment problem is solved at every stage; whenever subloops occur, one of them, say of length K, is chosen and a multiple branch into K subproblems is made restricting in turn each of the variables that form the subloop to take a zero value.

Shapiro (56) modified Eastman's algorithm to eliminate two-city subtours. In Held and Karp's algorithm (26) the minimum spanning tree is used to derive a tighter bound thus creating a compact tree at the expense of heavier computation at each node.

The various branch-and-bound algorithms are highly successful, but it becomes evident that the computational work grows exponentially with the number of cities; thus there are bounds on the size of the problems solvable with these methods.

Hong (30) seems to be the first one to have recovered the appeal of the cutting planes approach. He automated some of the interactive parts of the Dantzig, et al. procedure and incorporated further cutting planes. He reported good results on moderately sized problems. Miliotis (40) combined Dantzig's linear programming idea with branch-and-bound techniques.

In order to improve the goodness of the cutting plane method, intensive studies of the facet-structure of

polytopes related to TSP were carried out, and tremendously large classes of inequalities, in addition to the subtour elimination constraints, essential for characterization of these polytopes were discovered. The most popular ones include:

1. 2-matching constraints, founded by Edmonds (15);
2. Comb inequality of Chvatal (7);
3. Generalized comb inequality, found by Grotschel and Padberg (23).

Based on the result of polytope research associated with symmetric traveling saleman problem, a number of cutting plane algorithms were developed.

Two algorithms using cutting planes were developed for solving the traveling saleman problem by Miliotis (41). In both algorithms the problem is started with a subset of constraints that define the problem. However, the two algorithms differ in the order in which the omitted constraints and the cutting planes that are required are generated. The computational time was reported to be far less than branch-and-bound method. However, in one tested problem the cutting planes failed to prove infeasibility in both cases. Grotschel (22) demonstrated how the knowledge of the facets of the polytope can be utilized to solve large scale traveling salesman problems. The facitial cutting planes were added in an interactive way.

Padberg and Hong (47) developed a quite sophisticated cutting planes algorithm which was tested to support the hypothesis that inequalities defining facets of the convex hull of tours are of substantial value in the solution of large scale traveling salesman problems. They also citied that the difficulty with the implementation of this approach lies in the identification of a suitable cutting plane, a problem that has been neglected in the work on the facial structure of polyhedra associated with various combinatorial problems of practical interest. Their method can obtain a tighter lower bound on the minimum tour if an optional tour can not be found; this occurs when identification of the cutting plane is not successful.

Crowder and Padberg (10) added one more stage to the Padberg and Hong cutting plane algorithm. In stage II, ordinary branch-and-bound is applied to the stage I result. If the stage II output is a tour, the process stops. Otherwise, it is a system of subtours. A new cut is generated from this system, and stage II is repeated. This is so far the most efficient algorithm for obtaining optimal solutions to the traveling saleman problem. However, it still requires about 50 minutes of IBM 370/168 CPU time for a 318-city problem.

Heuristics

Many heuristic procedures have been developed in order to be able to "solve" traveling salesman problem with large number of cities and produce answers which are near optimal. A complete description and comparison of these methods can be found in references 5, 14, 17 and 57.

Golden, et al. (18) classified the heuristics into three broad classes:

1. Tour construction procedures;
2. Tour improvement procedures;
3. Composite procedures.

Tour construction procedures generate an approximately optimal tour from the cost matrix. Golden & Stewart (19) citied three components which serve as key ingredients in tour construction procedures:

1. The choice of an initial subtour (or starting point);
2. The selection criteria;
3. The insertion criteria.

Generally speaking, this class of heuristics include:

1. Nearest neighbor method;
2. Clark and Wright's savings method;
3. Nearest insertion method;
4. Cheapest insertion method;
5. Arbitrary insertion method;

6. Farthest insertion method;
7. Convex hull insertion method;
8. Christifide's method.

Among them, nearest neighbor method is the most widely used heuristic in industry. The procedure can be summarized as follows:

Step 1: Start with any node as the beginning of a path.

Step 2: Find the node closed to the last node added to the path. Add this node to the path.

Step 3: Repeat step 2 until all nodes are contained in the path. Then join the last and the first nodes.

Farthest insertion method is described below to show the picture of the insertion methods.

Step 1: Start with a subgraph consisting of node i only.

Step 2: Find node K such that C (i,k)is maximal and form subtour i-k-i.

Step 3: Selection step. Given a subtour, find node k not in the subtour farthest from any node in the subtour.

Step 4: Insertion step. Find the link (i, j) in the subtour which minimizes C(i,k)+C(k,j)-C(i,j) and then insert node k between node i and node j.

Step 5: Go to step 3 unless a complete tour is obtained.

Tour improvement procedures start with a feasible solution, then use an edge exchange technique to improve the solution iteratively until no further improvement can be made.

Croes (9) is the first person who proposed the edge exchange technique called the inversion method. The basic idea is to iteratively change 2 links of an tour with 2 other links not inside the tour until no further improvement can be made.

Croes' heuristic can be depicted as follows:

Step 1: Find a trial solution to be the candidate tour.

Step 2: Form the slant matrix by rearranging the rows and columns of the cost matrix such that the rows and columns are placed consecutively according to the sequence order of the candidate tour. The corresponding cost of the candidate tour will then become the slant of the new matrix.

Step 3: Form the inversion savings matrix from the slant matrix. An inversion reverses the order of a partial sequence in the candidate tour.

Step 4: Identify as many allowable simultaneous cost saving inversions as possible.

Step 5: If the number of inversions is 0, then stop.

Step 6: Carry out the identified inversions and make the resultant tour as the candidate tour. Go to step 2.

Lin (36) formally proposed the edge-exchange technique and defined k-optimality of the tour as follow:

A tour is said to be k-optimal (k-opt) if it is impossible to obtain a tour with smaller cost by replacing k of its links by any other set of k links not in the tour. Lin also citied that Croes' inversion method actually obtained a 2-opt tour. He also proposed his 3-opt algorithm which would obtain a solution at least no worse than that of Croes' inversion method.

The 3-opt algorithm can be summarized below:

Step 1: Generate a trial solution as the candidate tour.

Step 2: Let Count=0. Count is the number of rotations the current candidate has experienced. When a rotation is exercised, the city in the last node is moved to the first node, all the other nodes are moved to their corresponding next nodes.

Step 3: Let Count=Count+1

Step 4: Let K=0. K is the number of nodes, starting from current beginning node, to be included in the

partial sequence which may be inverted or inserted into other link of the candidate tour.

Step 5: Let K=K+1

Step 6: Let J=K. J is the starting node of the link which may be broken in in order to accomodate the partial sequence identified in step 4.

Step 7: Let J=J+1

Step 8: Determine the new format of the partial sequence by comparing the potential cost of inversion and insertion.

Step 9: Determine whether exchange can generate cost savings. If yes, go to step 14.

Step 10: If J is less than number of Nodes-1, go to step 7.

Step 11: If K is less than number of Nodes-3, go to step 5.

Step 12: If Count=Nodes, then stop.

Step 13: Rotate the tour. Go to step 3.

Step 14: Carry out the exchanges and obtain the new tour (make the city in node J+2 be the first node). Go to step 2.

Perhaps the most powerful heuristic for traveling salesman problem is the one proposed by Lin and Kerninghan (37). Lin and Kerninghan's algorithm is a variable K-opt method which determines at each iteration how many edges to exchange and usually results in almost-optimal solution. However, it requires considerably more effort to code than 2-opt or 3-opt algorithm. Consequently, this algorithm is mainly used when tighter upper bounds are required as input to an optimal TSP procedure.

Golden, et al. (18) designed a class of heuristic methods which are computationally less expensive but are as accurate as the repeated use of the 3-opt procedure. These heuristics are composite procedures.

The basic structure of the composite procedures can be started as follows:

Stage 1: Obtain an initial tour using one of the tour construction procedures.

Stage 2: Apply 2-opt procedure and/or 3-opt procedure on the initial tour to obtain a near-optimal solution.

They made an extensive computer expermentation on the performance of different heuristics for different problem sizes. The results can be summarized as follows:

1. For smaller problems, all heuristics work well.
2. For 100-node problems, the tour construction procedures have difficulty getting within 3% of the best known solution. The nearest neighbor method, nearest insertion, and cheapest insertion procedure performed quite poorly in all cases, about 15-20% above best known solutions. The Clarke-Wright, farthest insertion, and arbitrary insertion algorithms worked surprisingly well, about 3-8% above best known solution. The composite procedures tested all worked very well and usually came within 3% of the best known solution.

CHAPTER III

PROBLEM ANALYSIS

In order to develop solution methods for the assembly planning problem, this research starts with an analysis of the nature of the problem. A complete analysis of the simultaneous operation processes of the robot, X-Y Table movements, and the feeder carrier is needed in order to have a thorough investigation of the complicated assembly planning problem in the robotized pcb assembly center. This is caused by the fact that the optimal assembly plan may largely depend on the coordination of these items. Figure 2 gives an example of a simultaneous operation processes chart for this type of assembly center.

It is assumed that at time 0 the robot is located at place point, the X-Y table is located at home position such that the last component location in the PCB is positioned at place point and the feeder carrier is located at its own home position which is some point halfway between the two extreme feeder locations.

Two types of delay may occur in the robotized PCB assembly center. These are:

1. Board delay (place delay), the delay time period during which the robot is forced idle at the place

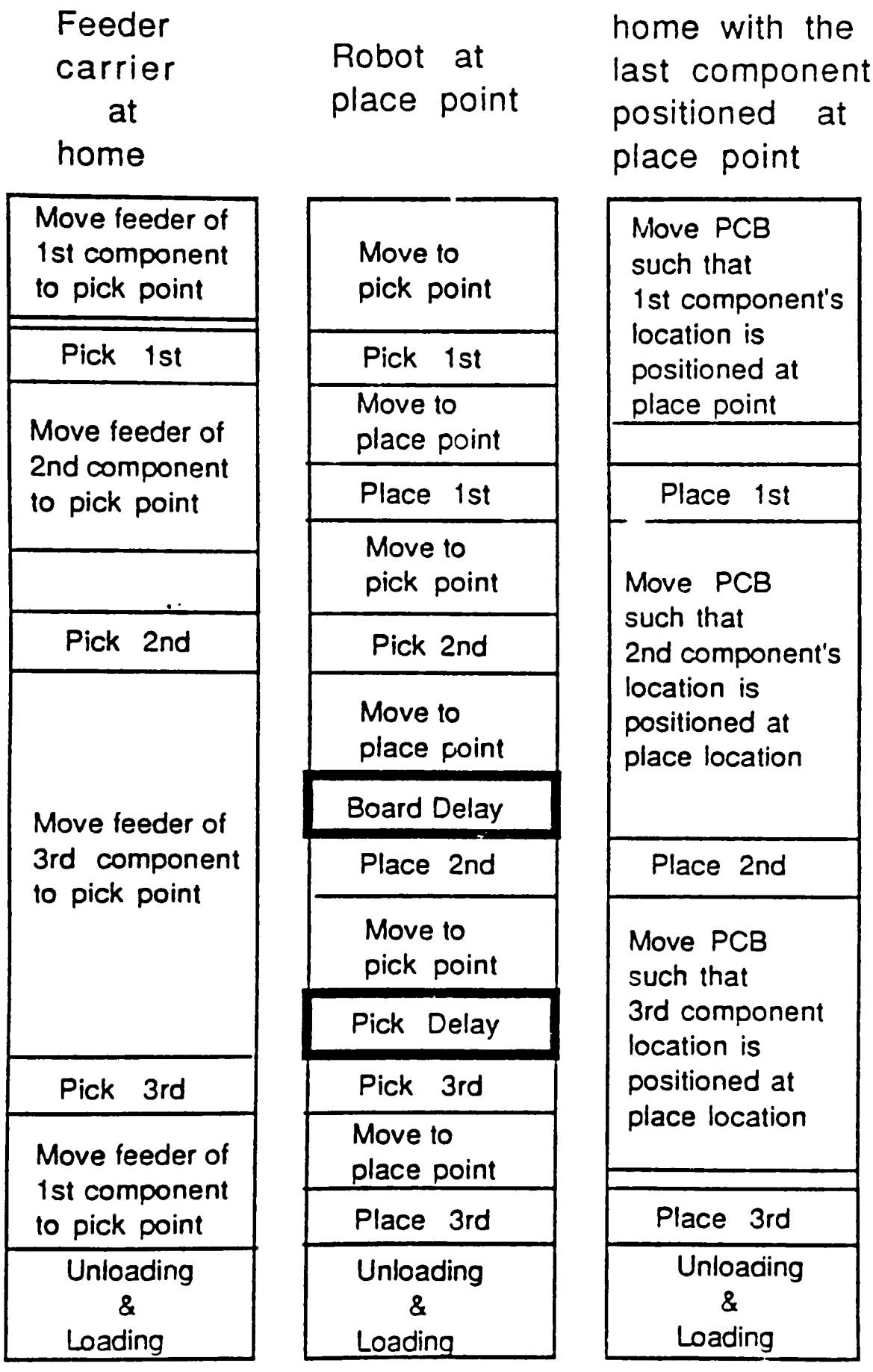


Figure 2: Simultaneous Operation Processes Chart

point waiting for the board to move the correct component location under the place point.

2. Pick delay (feeder delivery delay), the delay time period during which the robot is forced idle at the pick point waiting for the correct feeder to be delivered to the fixed pick point.

From Figure 2, we can know that the occurrance of board delay depends primarily on the relationship of the robot traveling time (from place point to pick point, picks up one component and back to place point) and the X-Y table travel time of the assembly step of interest. If this robot travel time is greater than the required X-Y table travel time, then no board delay will occur in this assembly step.

On the other hand, the occurrance of pick delay depends primarily upon the relationship of the robot travel time (travel from pick point to place point, place one component on PCB and travel back to pick point) and the delivery time required for the assembly step of interest. If the robot travel time is greater than or equal to the required delivery time, then no pick delay will occur.

The complexity of the mathematical models and the heuristic methods may be reduced if one or more types of delay do not occur. It is useful to classify the assembly environment into cases in which certain types of delay are

known to be nonexistent. Constructing mathematical models and heuristic methods according to performance characteristics of different assembly environments may help to avoid unnecessary computation efforts.

System Parameters

The assembly environment can be described by a set of parameters. These parameters can be classified into non-timing parmeters, timing parameters and a priority parameter. Non-timing parameters include:

1. N: number of components in PCB to be assembled by the PCB assembly center.
2. NTT: number of component types in PCB to be assembled by the PCB assembly center (NTT $\leq$ N)
3. NF: number of available feeders in the feeder carrier

Priority parameter (HPO) is used to indicate the form of the assembly problem. HPO=A means the assembly planning problem is of form A. That is, it is desired to minimize total assembly cycle time first and then to minimize total X-Y table travel time. HPO=B means the assembly planning problem is of form B. That is, it is desired to minimize total X-Y table travel time first and then minimize total assembly cycle time.

Timing parameters which have the potential to influence board delay and pick delay include the following which then will be defined in subsequent paragraphs.

1. Accuracy capability of the X-Y table (AC);
2. Robot normal round trip time (RT);
3. Board extreme moving time (BE);
4. Unit delivery time of the feeder carrier (UT);
5. Delivery capability of the feeder carrier (DC);
6. Shortest placing related timing parameter (SPT).

In the electronic industry, one of the things of the most concern is quality of product. The quality of a PCB may be affected by many variables in different stages of the entire production cycle. In PCB assembly, the quality of the PCB is very much dependent upon the accuracy with which the components are placed on their prespecified location points on the PCB. In the PCB assembly center considered in this research, the placement accuracy is largely dependent on the X-Y table movements. For a single X-Y table movement, the longer the X-Y table travels the less placement accuracy the X-Y table can keep. Thus, accuracy capability (AC) can be defined as the maximum travel time of a single X-Y table movement without causing any inaccurate placement.

Robot normal round trip time (RT) consists of the time for traveling from pick point to place point, placing component on the PCB and then traveling back to pick point without board delay. Alternatively, robot normal round trip time can be defined as the robot travel time from place point to pick point, picking up a component and traveling back to place point without pick delay.

Board extreme moving time (BE) is the time needed for the X-Y table to move the distance between the two extreme points of the PCB under assembly.

Unit delivery time of the feeder carrier (UT) is the time needed for the feeder carrier to move the distance between two neighboring feeders.

Delivery capability of the feeder carrier (DC) is defined as Int(RT/UT). In effect, it is the maximum number of units the feeder carrier can move without causing pick delay during robot normal round trip time.

If the delivery capability of the feeder carrier (DC) is less than one, i.e., the unit delivery time of the feeder carrier (UT) is greater than robot normal round trip time, then pick delay can not be avoided. This will occur when robot speed is very fast and/or when the feeder carrier is quite slow and seems to suggest a mismatch in the application of the feeder carrier.

Shortest placing related timing parameter (SPT) is the minimum of the three placing related parameters, BE, AC, and RT.

Depending on the relationship of the non-timing parameters, the PCB assembly environment can be classified into two different cases. These are:

1. N≤NF: Where the number of components in the PCB is less than or equal to the number of feeders in the feeder carrier.
2. N>NF: Where the number of components in the PCB is greater than the number of feeders in the feeder carrier.

On the other hand, depending on the relationship of the timing parameters, the PCB assembly environment can be classified into three different cases. These are:

1. SPT=BE
2. SPT=AC
3. SPT=RT

Combining the two forms of the assembly planning problem, the PCB environment can be classified into 12 classes. These are:

1. (HPO=A/N≤NF/SPT=BE)
2. (HPO=A/N≤NF/SPT=AC)
3. (HPO=A/N≤NF/SPT=RT)

4. (HPO=B/N≤NF/SPT=BE)
5. (HPO=B/N≤NF/SPT=AC)
6. (HPO=B/N≤NF/SPT=RT)
7. (HPO=A/N>NF/SPT=BE)
8. (HPO=A/N>NF/SPT=AC)
9. (HPO=A/N>NF/SPT=RT)
10. (HPO=B/N>NF/SPT=BE)
11. (HPO=B/N>NF/SPT=AC)
12. (HPO=B/N>NF/SPT=RT)

Interpretation of these assembly environment classes is rather straight forward. For example, the first assembly environment class may be interpreted as follows:

1. The assembly planning problem is of form A. That is, the priority order of the two objectives is to minimize total assembly cycle time first and then to minimize total X-Y table travel time.
2. The number of components in the PCB is less than or equal to the number of feeders in the feeder carrier.
3. Board extreme moving time (BE) is the smallest among the three placing related timing parameters. That is, the time needed for the X-Y table to move the distance between two extreme points of the board is no greater than the robot normal round trip time (RT). Moreover, board extreme moving time (BE) does

not exceed the accuracy capability of the feeder carrier (RT).

In this research, a symbol of '*' is used to represent a group of assembly environment classes. For example, (HPO=A/N>NF/*) represents the group of assembly environment classes in which the assembly planning problem is of form A and the number of components in the PCB is greater than the number of feeders in the feeder carrier.

Analysis of Pick Delay

One of the objectives in the assembly planning problem is to minimize total assembly cycle time. For the PCB assembly center considered in this research, the cycle time may be affected by two types of delay: pick delay and board delay.

It is helpful to have an analysis of the factors which may cause the two types of delay such that methods avoiding the two types of delay may be derived.

Pick delay may be affected by the following factors:

1. The delivery capability of the feeder carrier;
2. The assignment of components to the feeders;
3. The assembly sequence used to assemble the PCB;
4. The interference from the board delay.

If the speed of feeder carrier is sufficiently fast that during robot normal round trip time the feeder carrier can move any feeder to pick point in time, then no pick delay will happen no matter what assignment of feeders is selected. However, the speed of the feeder carrier is usually slow such that the above condition will not occur. Therefore, the assignment of feeders becomes important in obtaining zero pick delay for a selected assembly sequence.

For assembly environment classes in which the number of components in the PCB is less than or equal to the number of available feeders (*/N≤NF/*), two assignment methods can be used.

One way of doing it is to assign components to the feeders in the order of assembly sequence. By this sequential assignment, the delivery of the component of the first feeder to the pick point requires (N-1)*UT amount of time if originally the last feeder is under the pick point. All other delivery times are UT. Therefore, pick delay will not occur if the following constraints are satisfied:

1. Delivery capability is greater than or equal to 1 .
2. The feeder carrier is allowed to move during unloading/loading time period.
3. The summation of unloading/loading time and the robot normal round trip time is greater than or equal to (N-1)*UT.

For the robotized PCB assembly center considered in this research, the unloading/loading time can not be utilized to compensate for feeder carrier's movement. Thus, the sequential assignment may not guarantee zero pick delay for this robotized PCB assembly center.

Another assignment method is suggested in this research to guarantee zero pick delay without making use of unloading/loading time. This method, called Jumping Assignment Method, makes use of consecutive N feeders out of the NF feeders in the feeder carrier. The remaining NF-N feeders are treated as nonexistent. This assignment method assigns the first component in the assembly sequence to the first feeder. The assignment method will then make a two-step jump to the third feeder location and assign the second component to the third feeder. After then, a number of two-step will be made, toward the same direction, until a two-step jump can not be made and the destinations of these two-step jumps will be assigned to hold the components in the order of the remaining components in the sequence. Whenever a two-step jump can not be made toward the same direction, an one-step jump is made to reach the last feeder. When the last feeder is assigned a component, the next jump changes direction. If the previous jump is a one-step jump, the jump from the last feeder is a two-step jump.

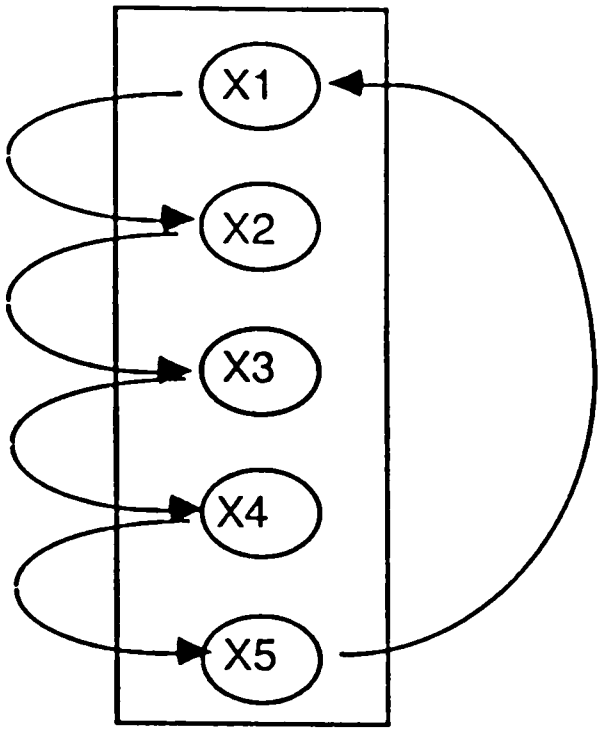
Otherwise, the jump from the last feeder is a one-step jump. After that, a number of two-step jumps are made until all components are assigned.

An example of assigning five components in an assembly sequence X1, X2, X3, X4 and X5 to five feeders in the feeder carrier by sequential assignment method is presented in Figure 3 (a) in which component X1 is assigned to the first feeder, X2 is assigned to the second, X3 is assigned to the third, X4 is assigned to the fourth, and X5 is assigned to the fifth. Figure 3 (b) shows the use of jumping assignment method in which component X1 is assigned to the first feeder, X2 is assigned to the third, X3 is assigned to the fifth, X4 is assigned to the fourth, and X5 is assigned to the second.

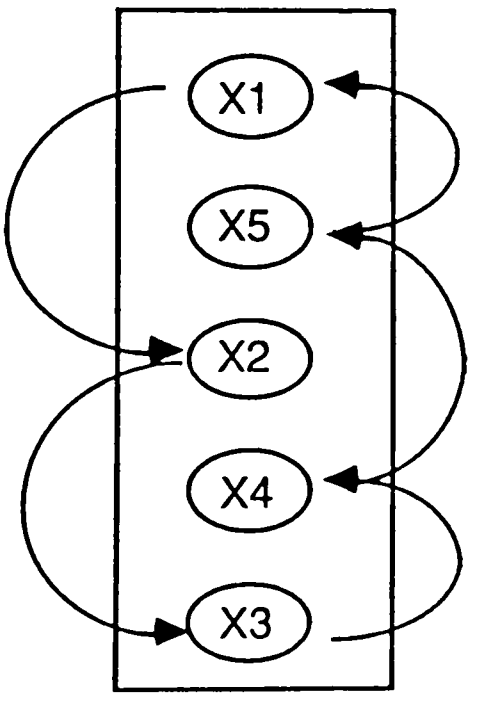
Under the condition that unloading/loading time can not be utilized to compensate for the feeder carrier's movement, it can be easily seen that zero pick delay can be guaranteed by application of the Jumping Assignment Method if DC>2.

In summary, the following statement will hold:

Statement 1: If $N \leq NF$ and $DC \geq 2$, then zero pick delay is guaranteed by applying the Jumping Assignment Method for every assembly sequence selected.



(a) Sequential Assignment Method



(b) Jumping Assignment Method

Figure 3: Sequential Assignment Method and Jumping Assignment Method

Analysis of Board Delay

Board delay may be affected by the following factors:

1. The magnitude relationship of the three placing related timing parameters: BE, RT and AC.
2. The assembly sequence used to assemble the PCB.
3. The influence of pick delay.

A set of analysis diagrams is provided in this section to facilitate understanding of the analysis of board delay. Since board delay deals with time units, the X-Y coordinate system of these diagrams is choosen to be in time units. That is, the length between any two points in these diagrams represents the travel time needed between the two points.

Graphical representations of the three placing related timing parameters, BE, AC and RT their relationships with X-Y table travel time of a single assembly step (OB(i,j)) are introduced first. Given a rectangular PCB, the extreme distance of the PCB is always the distance between the two end points of the diagonal. Thus, the board extreme moving time (BE) is the time required by the X-Y table to move the board between the two end points of the diagonal and can be represented by the length of segment BE in Figure 4

The assembly steps can be represented by arrows as shown in Figure 4 The tail of an arrow represents the location of the starting component of the assembly step and

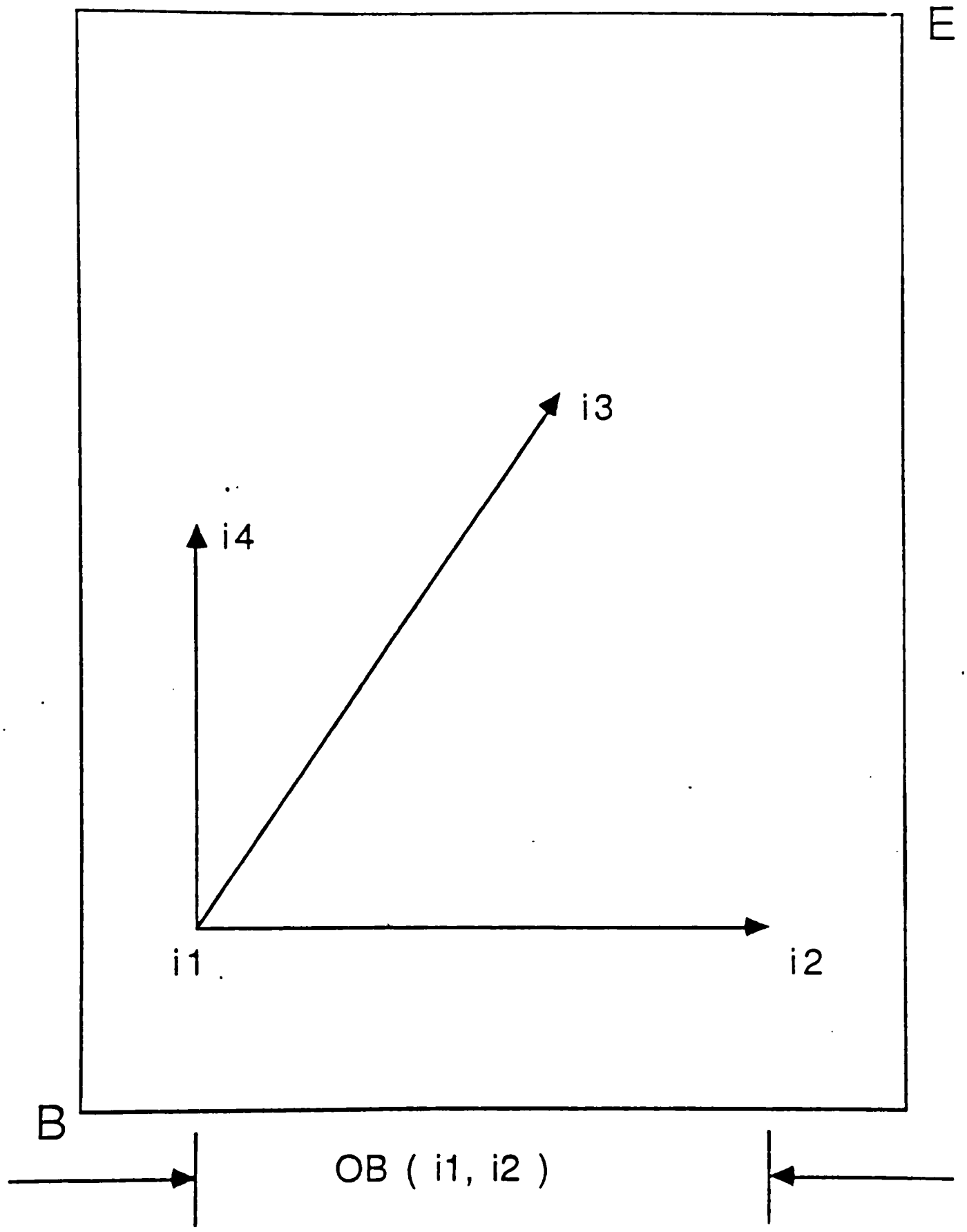


Figure 4: Graphical Representation of Assembly Steps

the head of an arrow represents the location of the ending component of the assembly step. The X-Y table travel time of an assembly step, OB(i,j), is represented by the length of the arrow. The set of arrows with a common arrow tail represents the potential assembly steps starting from the same component.

By definition, the following relationship will always hold:

$BE \geq OB(i,j)$ for any assembly step (i,j)

This relationship can be represented by Figure 5 A circle of radius BE will contain every point on the PCB given the center of the circle is on the PCB. Since all the possible assembly steps represented by the arrows are inside the rectangle, there is no doubt that all the assembly steps must be inside these circles. Thus, given a circle which has radius BE and centers at a point on the rectangular PCB, no assembly steps starting from the centering point can exceed the radius of the circle.

Starting from a component location, an assembly step will generate inaccurate placement if the required X-Y table travel time is greater than the accuracy capability of the X-Y table (AC). These infeasible assembly steps must be excluded from the assembly sequence. Given the location of a component, a circle with radius AC can be established to

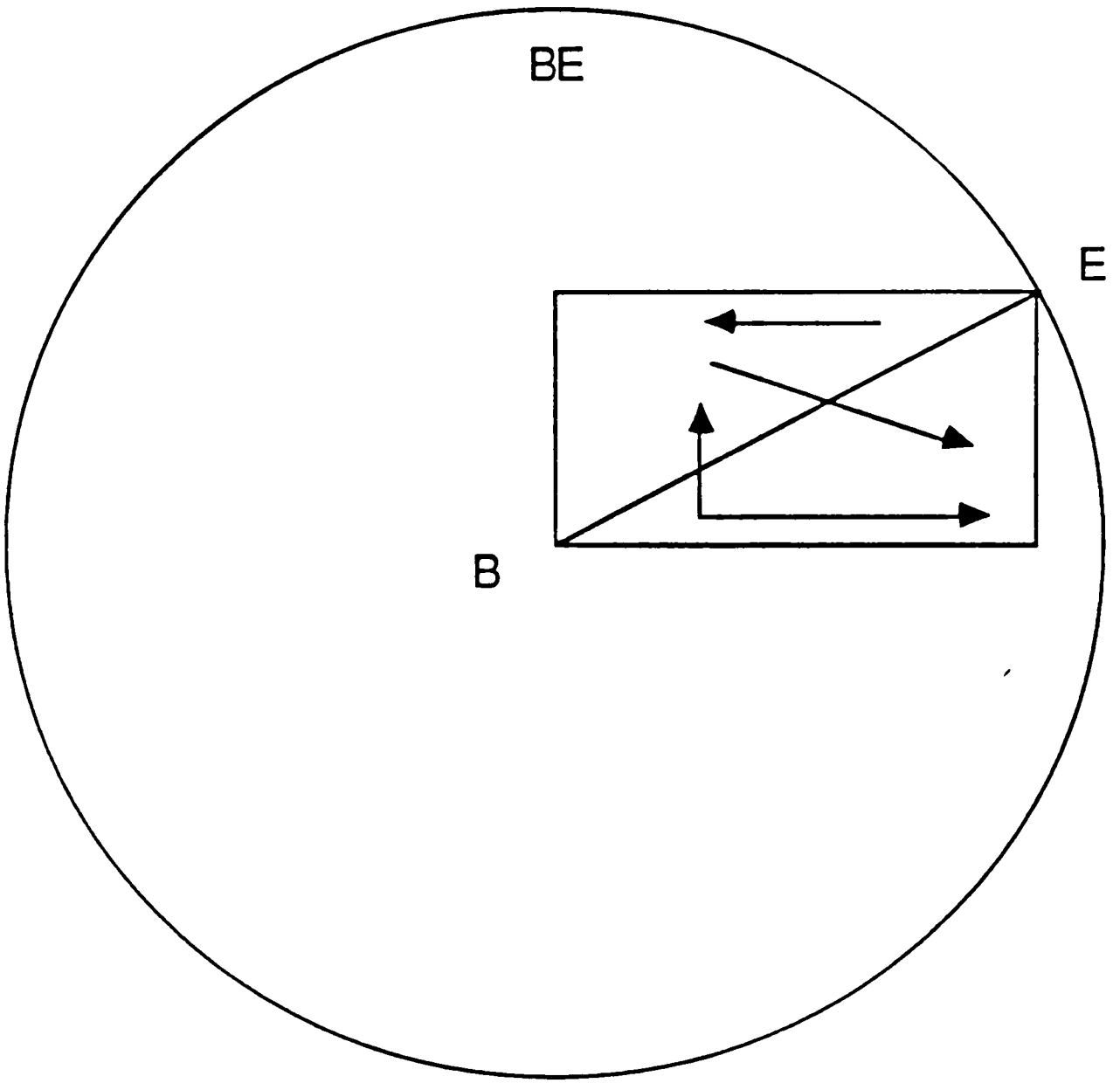


Figure 5: Graphical Representation of BE and Potential Assembly Steps

determine which assembly steps, starting from the center of the circle, are feasible assembly steps and which are infeasible.

For example, when a component, say il, is placed on the PCB, it is of interest to determine which assembly steps, starting from component il, are feasible assembly steps and which are infeasible assembly steps. A circle with radius AC and center point il can be established as shown in Figure 6 The assembly steps of interest are those arrows with tail at the center point. Feasible assembly steps starting with component il are those arrows with heads inside circle AC because their corresponding X-Y table travel times are no greater than AC. Infeasible assembly steps are those arrows with heads outside circle AC. Thus, given a circle with radius AC, feasible assembly steps starting from the center point of the circle are inside the circle while infeasible assembly steps have arrow heads outside the circle.

From analysis of the simultaneous operation processes chart, the robot would require at least RT time units to return back to the place point after one component is placed on the PCB. Meanwhile, the X-Y table is moving the board to the location of the next component. If the corresponding X-Y table travel time is greater than the robot normal round trip time (RT), this assembly step will generate board delay. Otherwise, board delay will not occur.

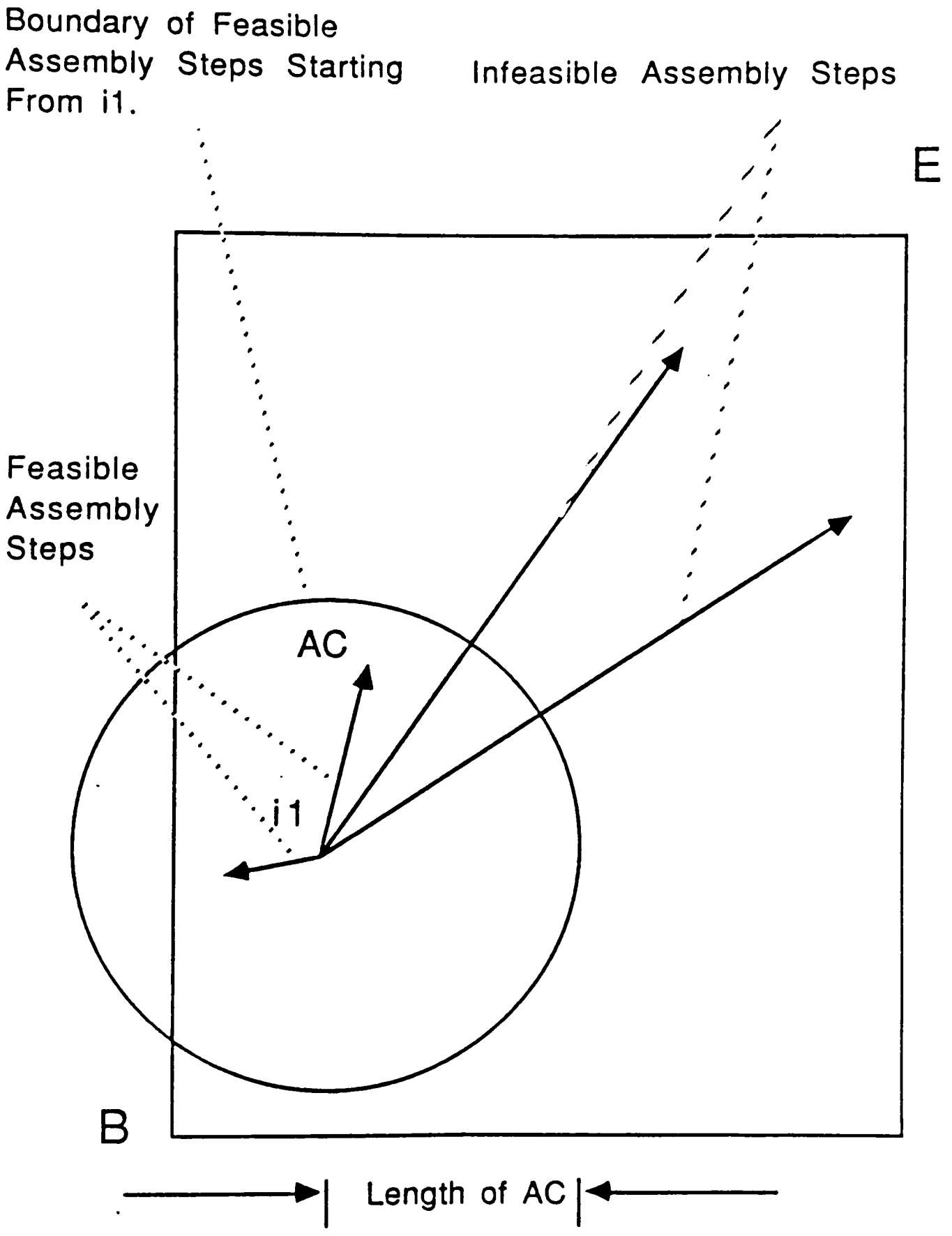


Figure 6: Feasible Assembly Steps and Infeasible Assembly Steps

When a component, say il, is placed on the PCB, a circle with radius RT and center point il can be established as shown in Figure 7 Circle RT means that the X-Y table can move the boards to any point within the circle having its center at il. Arrows with tail in the center point il and head outside circle RT represent the assembly steps which would introduce board delay. Arrows with tail in the center point il and head inside circle RT represent the assembly steps which would not introduce board delay when included in the assembly sequence.

There are three possible magnitude relationships of the three placing related timing parameters: (1) SPT=BE, (2) SPT=AC, and (3) SPT=RT.

SPT equals BE if one of the following two relationships hold: (1) ($RT \geq AC \geq BE$); or (2) ($AC \geq RT \geq BE$). Let OB(i,j) be the corresponding board travel time for assembly step (i,j) where component j is placed after component i is placed on the PCB. It is clear that $RT > BE$ and $AC \geq BE$ in both cases. By definition, $BE \geq OB(i,j)$ for any assembly step (i,j). Therefore, $RT \geq OB(i,j)$ for any assembly step (i,j) and $AC \geq OB(i,j)$ for any assembly step (i,j). That means during robot normal round trip time period no board delay will occur no matter what assembly step is selected. Also, no assembly step will generate inaccurate placement.

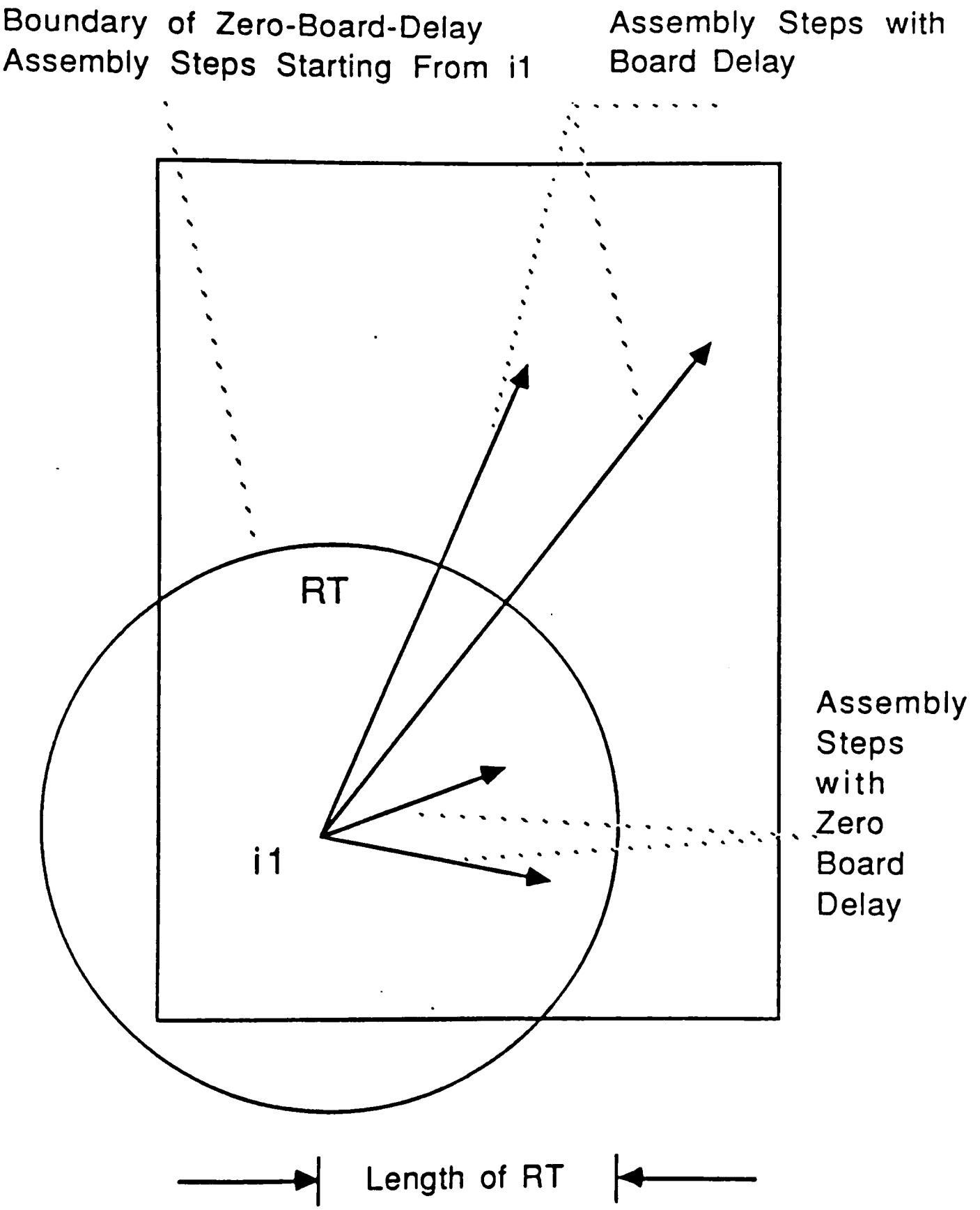


Figure 7: Asssembly Steps with Board Delay and Assembly Steps without Board Delay

The relationship (RT≥AC≥BE) may be graphically represented by Figure 8 while the relationship (AC≥RT≥BE) may be represented by Figure 9

Both figures contain a rectangle and three concentric circles with radii RT, AC and BE. The rectangle represents the PCB of interest with board extreme moving time BE. The three concentric circles have the same center point at the location of an arbitrarily selected component. It can be seen that all the assembly steps starting with component il lie inside circle AC. As explained in Figure 6, no assembly steps starting from component il would cause infeasible placement. Meanwhile, all the assembly steps are inside circle RT. As explained in Figure 7, no assembly steps starting from component il would generate board delay. By establishing the 3 concentric circles on the location of each component, the following statement can hold:

Statement 2: If SPT=BE, then no assembly steps would result in infeasible placement and no board delay would occur no matter what assembly sequence is selected.

SPT equals AC if one of the following two relationships hold: (1) (RT≥BE≥AC) or (2) (BE≥RT≥AC). The relationship (RT≥BE≥AC) is graphical represented in Figure 10 while the relationship (BE≥BE≥AC) is represent in Figure 11 Starting

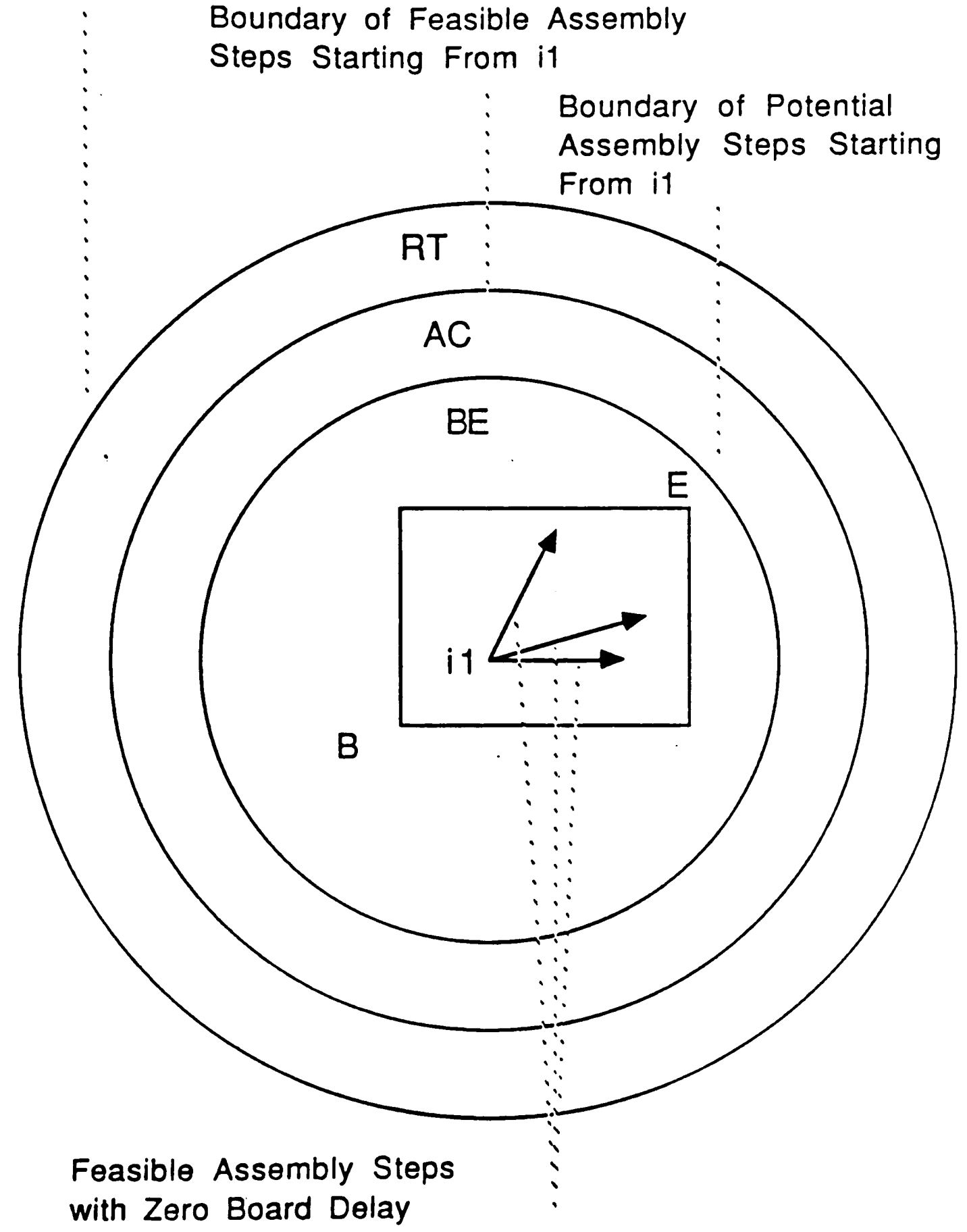


Figure 8: Graphical Representation of the Relationship $RT \geq AC \geq BE$

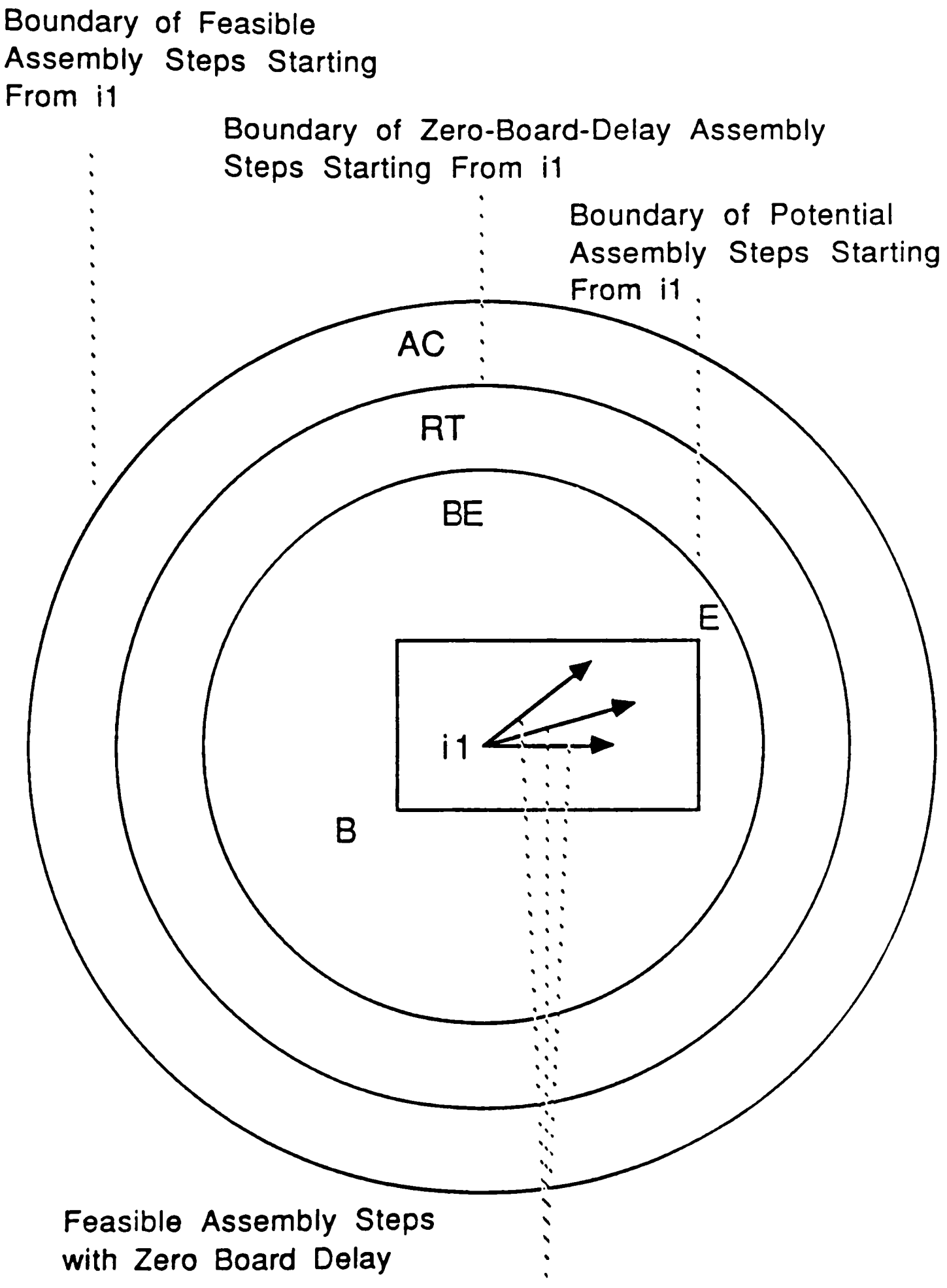


Figure 9: Graphical Representation of the Relationship $AC \geq RT \geq BE$

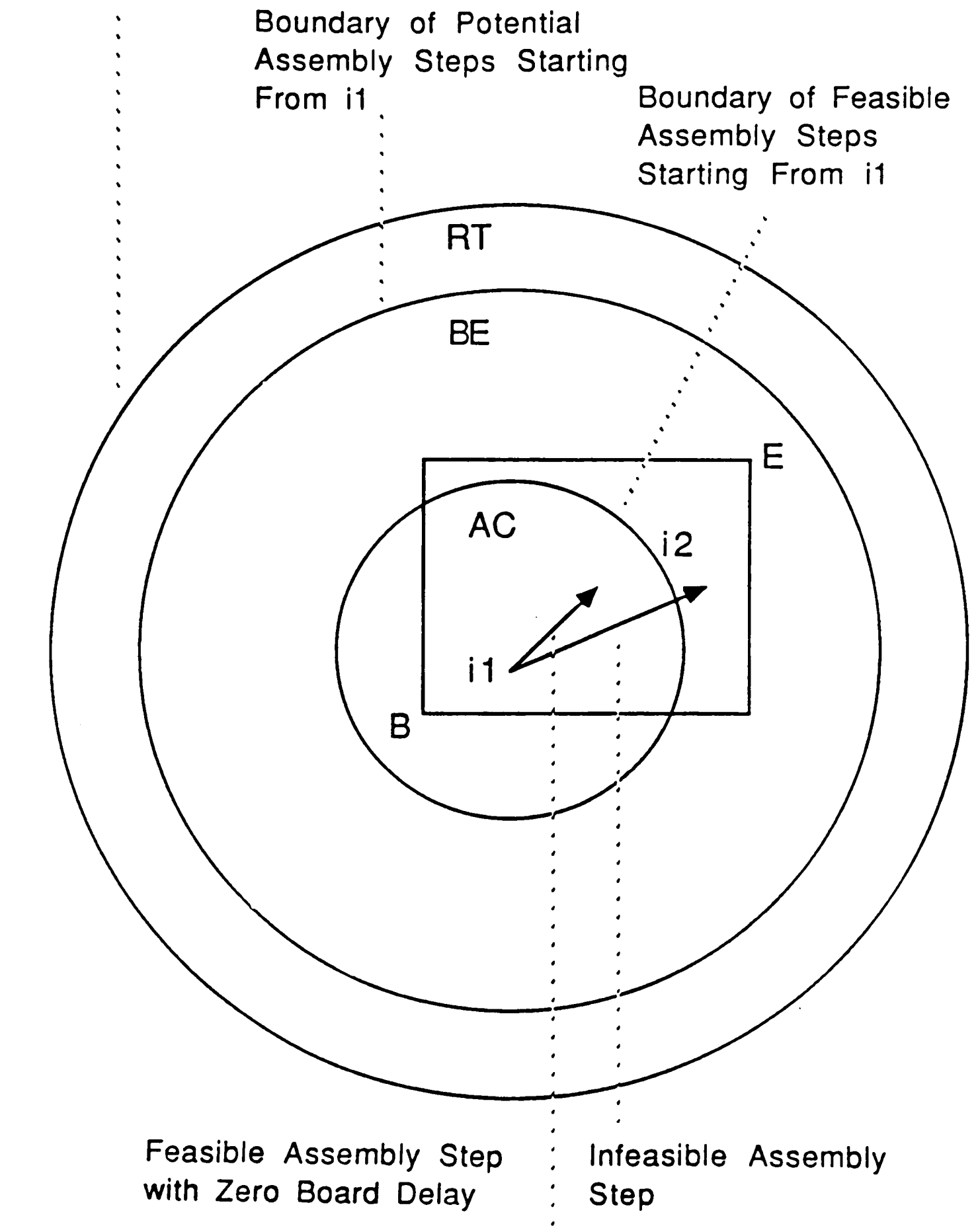


Figure 10: Graphical Representation of the Relationship $RT \geq BE \geq AC$

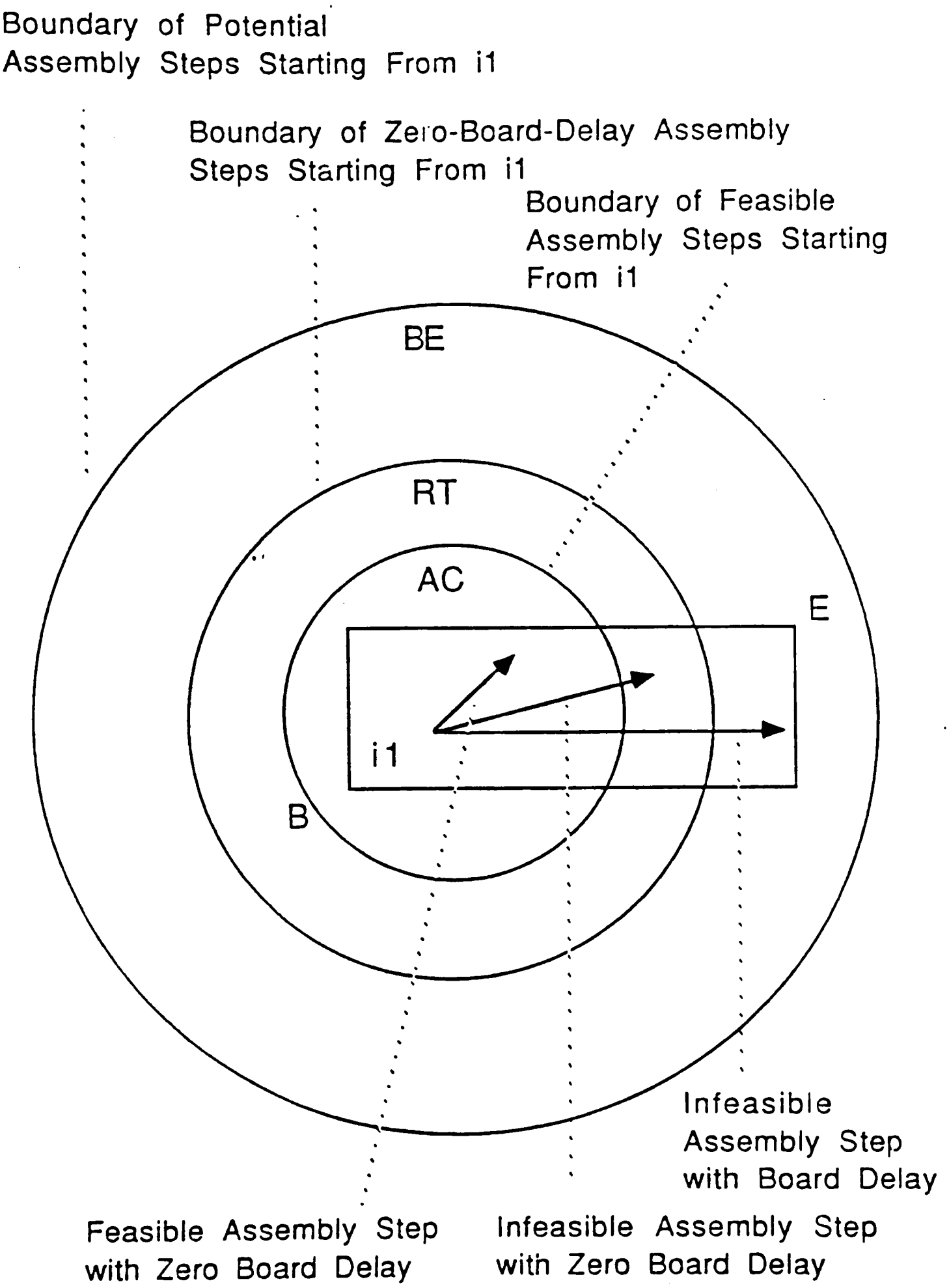


Figure 11: Graphical Representation of the Relationship BE≥RT≥AC

with a component, say il, some assembly steps are outside circle AC. That means some assembly steps starting from component il require X-Y table travel time greater than AC and would introduce infeasible placement. These infeasible assembly steps must be excluded from the assembly sequence. Once they are eliminated, the remaining assembly steps are inside circle AC and circle RT. The following statement will hold:

Statement 3: If SPT=AC, inaccurate placement might be introduced by some assembly steps. These infeasible assembly steps must be excluded from the assembly sequence. Once they are eliminated, the remaining feasible assembly steps will not generate any board delay.

SPT equals RT if one of the following two relationships hold: (1) (BE≥AC≥RT) or (2) (AC≥BE≥RT). The relationship (BE≥AC≥RT) is depicted in Figure 12 while the relationship (AC≥BE≥RT) is represented in Figure 13

From Figure 12, it is clear that circle AC is inside circle BE. That means some assembly steps starting from component il may introduce infeasible placement because some assembly sequences can contain steps with board travel that exceeds the limit due to a loss of accuracy in placement. These assembly steps must be excluded from the assembly

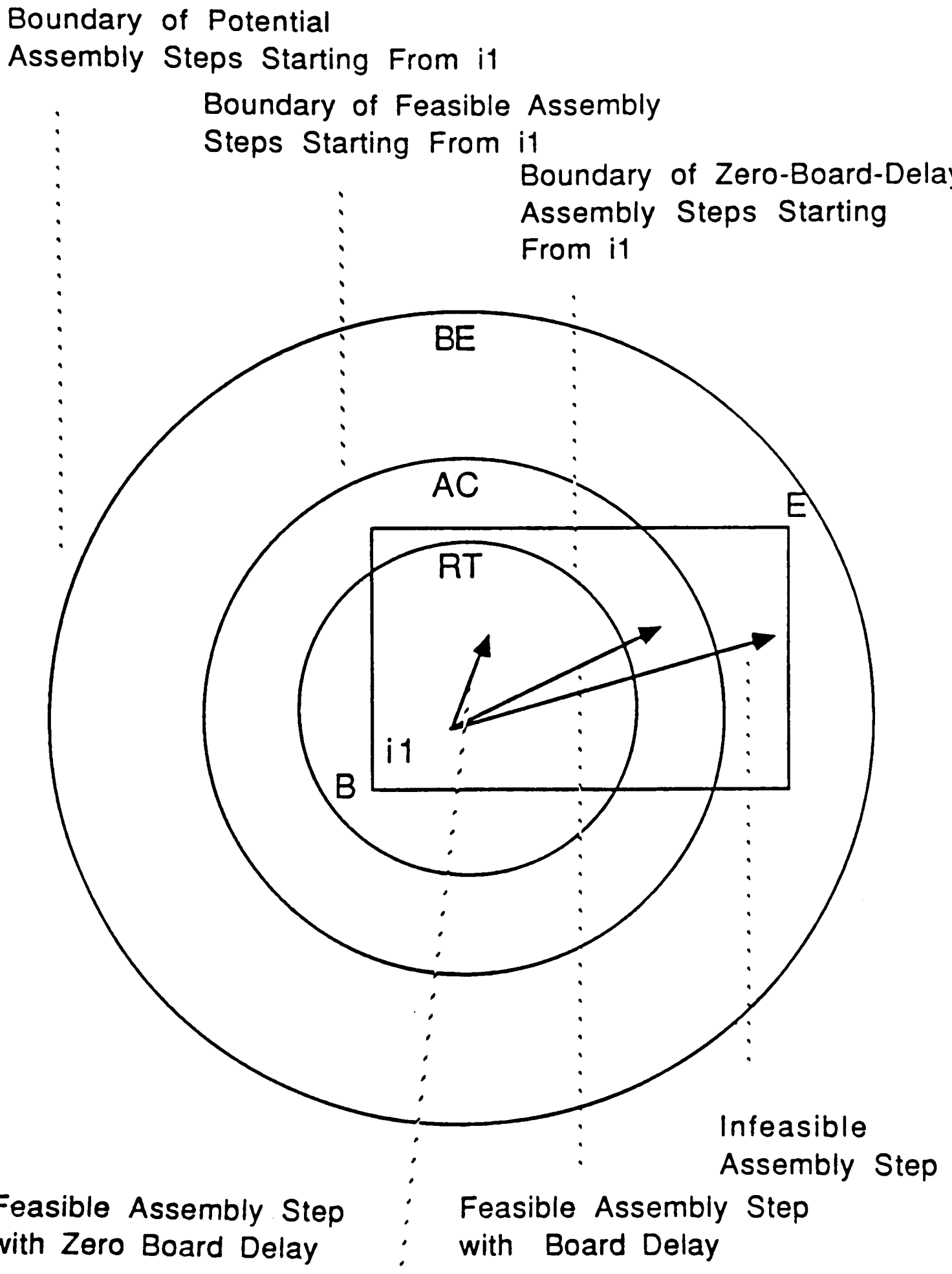


Figure 12: Graphical Representation of the Relationship BE≥AC≥RT

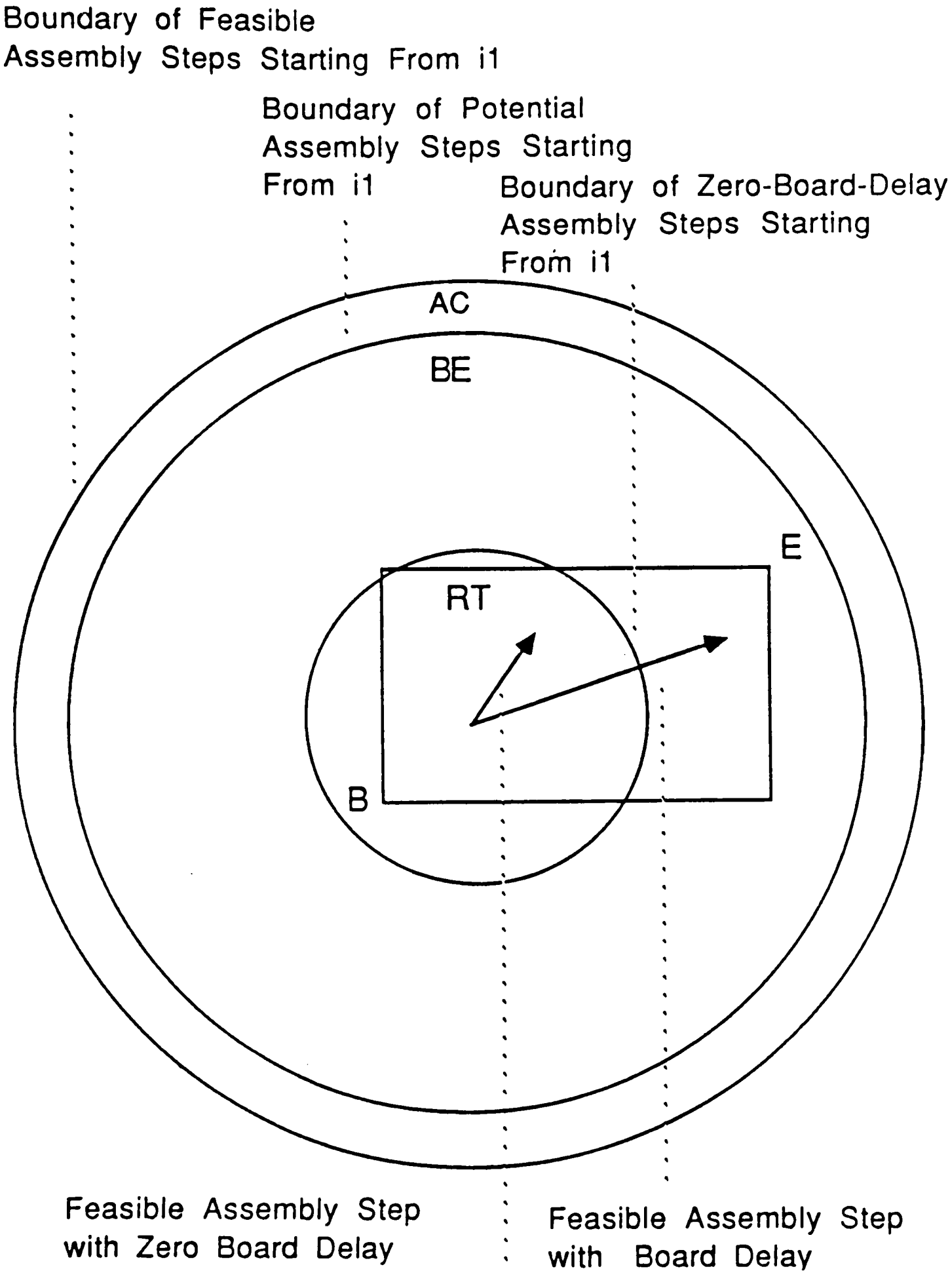


Figure 13: Graphical Representation of the Relationship $AC \geq BE \geq RT$

sequence. After all these infeasible assembly steps are excluded, the remaining feasible assembly steps lie inside circle AC. However, circle AC has a larger radius than Circle RT. Thus, some feasible assembly steps may cause board delay.

From Figure 13, Circle BE is inside circle AC. That means all the assembly steps are feasible. However, some of them are outside circle RT. Thus, some assembly steps may cause board delay. Thus, the following statement will hold:

Statement 4: If SPT=RT, zero board can not be guaranteed even though infeasible assembly steps are excluded from the assembly sequence.

However, when zero board delay is not guaranteed at SPT=RT, it is still possible to obtain it if a proper assembly sequence is selected.

Summary of Problem Analysis

As discussed previously, there may be 12 different assembly environment classes. From analysis of pick delay and board delay, it is clear that board delay will not occur and pick delay can be avoided no matter what assembly sequence is selected in assembly environment classes (HPO=A/N≤NF/SPT=BE) and (HPO=B/N≤NF/SPT=BE). The optimal assembly plans for these two enviroment classes become the

same. In both of them the assembly sequence with minimal X-Y table travel time is sought; then application of the Jumping Assignment Method is made with respect to the assembly sequence just obtained. (HPO=AB/N≤NF/SPT=BE) is used to represent these two assembly environment classes. For the same reason, (HPO=AB/N≤NF/SPT=AC) is used to represent the assembly environment classes (HPO=A/N≤NF/SPT=AC) and (HPO=B/N≤NF/SPT=AC). The definitions of the ten different assembly environment classes are presented in Table 1 Every class may have its own performance characteristics. The mathematical models as well as heuristic methods can be developed on the basis of those performance characteristics. Table 2 summarizes these performance characteristics for each class.

TABLE 1

Definitions of the 10 Assembly Environment Classes

(HPO=AB/N≤NF/SPT=BE):	(1) The priority order of the two objectives can be either to minimize total assembly cycle time first and then to minimixe total X-Y table travel time or to minimize total X-Y table travel time first and then to minimize total cycle time. (2) The number of components in the PCB is less than or equal to the number of feeders in the feeder carrier. (3) Board extreme moving time (BE) is the smallest among the three placing related timing parameters.
(HPO=AB/N≤NF/SPT=AC):	(1) The priority order of the two objectives can be either to minimize total assembly cycle time first and then to minimixe total X-Y table travel time or to minimize total X-Y table travel time first and then to minimize total cycle time. (2) The number of components in the PCB is less than or equal to the number of feeders in the feeder carrier. (3) Accuracy capability (AC) is the smallest among the three placing related timing parameters.
(HPO=A/N≤NF/SPT=RT):	(1) The assembly planning problem is of form A. It is desired to minimize total assembly cycle time first and then to minimize total X-Y table travel time. (2) The number of components in the PCB is less than or equal to the number of feeders in the feeder carrier. (3) Robot normal round trip time (RT) is the smallest among the three placing related timing parameters.
(HPO=B/N≤NF/SPT=RT):	(1) The assembly planning problem is of form B. It is desired to minimize total X-Y table travel time first and then to minimize total assembly cycle time. (2) The number of components in the PCB is less than or equal to the number of feeders in the feeder carrier. (3) Robot normal round trip time (RT) is the smallest among the three placing related timing parameters.

Table 1 - Continued

(HPO=A/N>NF/SPT=BE):	(1) The assembly planning problem is of form A. It is desired to minimize total assembly cycle time first and then to minimize total X-Y table travel time. (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier. (3) Board extreme moving time (BE) is the smallest among the three placing related timing parameters.
(HPO=A/N>NF/SPT=AC):	(1) The assembly planning problem is of form A. It is desired to minimize total assembly cycle time first and then to minimize total X-Y table travel time. (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier. (3) Accuracy capability (AC) is the smallest among the three placing related timing parameters.
(HPO=A/N>NF/SPT=RT):	(1) The assembly planning problem is of form A. It is desired to minimize total assembly cycle time first and then to minimize total X-Y table travel time. (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier. (3) Robot normal round trip time (RT) is the smallest among the three placing related timing parameters.
(HPO=B/N>NF/SPT=BE):	(1) The assembly planning problem is of form B. It is desired to minimize total X-Y table travel time first and then to minimize total assembly cycle time. (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier. (3) Board extreme moving time (BE) is the smallest among the three placing related timing parameters.
(HPO=B/N>NF/SPT=AC):	(1) The assembly planning problem is of form B. It is desired to minimize total X-Y table travel time first and then to minimize total assembly cycle time. (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier. (3) Accuracy capability (AC) is the smallest among the three placing related timing parameters.

Table 1 - Continued

(HPO=B/N>NF/SPT=RT):	(1) The assembly planning problem is of form B. It is desired to minimize total X-Y table travel time first and then to minimize total assembly cycle time. (2) The number of components in the PCB is greater than the number of feeders in the feeder carrier. (3) Robot normal round trip time (RT) is the smallest among the three placing related timing parameters.

TABLE 2

Summary of Performance Characteristics

Assembly Environment Class	Zero Pick Delay is Guaranteed	Zero Board Delay is Guaranteed
(HPO=AB/N≤NF/SPT=BE)	yes	yes
(HPO=AB/N≤NF/SPT=AC)	yes	yes*
(HPO=A/N≤NF/SPT=RT)	yes	no
(HPO=B/N≤NF/SPT=RT)	yes	no
(HPO=A/N>NF/SPT=BE)	yes**	yes
(HPO=A/N>NF/SPT=AC)	no	yes*
(HPO=A/N>NF/SPT=RT)	no	no
(HPO=B/N>NF/SPT=BE)	no	yes
(HPO=B/N>NF/SPT=AC)	no	yes*
(HPO=B/N>NF/SPT=RT)	no	no

* Infeasible assembly steps must be eliminated

** Components of the same type can be assembled consecutively. From the view point of picking operation, the assembly sequence is reduced to a more compact form in which component types are the members of the sequence, instead of the components. Jumping assignment method can be applied with respect to the compact form of the assembly sequence to guarantee zero pick delay.

CHAPTER IV

MATHEMATICAL MODELS

The optimal assembly plan seeks the determination of assembly sequence and assignment of components to the feeders to sequentially best satisfy the two prioritized objectives. The two objectives are to minimize total X-Y table travel time and to minimize total cycle time. The priority order of the two objectives may be different under different production situations. For the PCB assembly center considered in this research, the minimal assembly cycle time (MAC) without any delay can be expressed by the following equation:

$$MAC=N*PT+N*RT \tag{4.1}$$

where

N: the number of components to be placed in the PCB

PT: the time needed for robot to pick/place a single component

RT: robot normal round trip time

For a given assembly environment class, all the variables in Eq. (4.1) are known constants. Consequently, minimization of total assembly cycle time is equivalent to minimization of total delay. The mathematical models for different assembly environment classes are presented in the following sections.

(HPO=AB/N<NF/SPT=BE)

According to the performance characteristics identified in the previous chapter for this assembly environment class, no delay will occur no matter what assembly sequence is selected. Thus, the assembly planning problem can be solved by concentrating on one objective: minimize total X-Y table travel time. The mathematical model can be formulated as a traveling salesman problem as follows:

$$\text{MIN} \sum_{\text{all } (i,j)} OB(i,j)*X(i,j) \qquad (4.2)$$

subject to

$$\sum_{\text{all } i} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \qquad (4.3)$$

$$\sum_{\text{all } j} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \qquad (4.4)$$

$$U(i)-U(j)+N*X(i,j) < N-1 \qquad \text{for } i = 2,3,\ldots,N;\ j = 2,3,\ldots,N \qquad (4.5)$$

$$X(i,j) \text{ non-negative integer} \qquad \text{for all } (i,j) \qquad (4.6)$$

where

(i,j): the assembly step with component i immediately preceeding component j

OB(i,j): the X-Y table travel time required for assembly step (i,j)

$$X(i,j)=\begin{cases} 1, & \text{if assembly step } (i,j) \text{ is included in the assembly sequence.} \\ 0, & \text{otherwise.} \end{cases}$$

N: number of components in the PCB

Constraints(4.3), (4.4) and (4.6) limit each component to be placed on the PCB to exactly one time. Constraint (4.5) is called a subtour elimination constraint. The function of this constraint is to make the assembly sequence be a complete tour.

After the optimal assembly sequence is determined by the model, the Jumping Assignment Method is used with respect to the resultant assembly sequence to obtain the assignment of components to the feeders in the feeder carrier.

(HPO=AB/N≤NF/SPT=AC)

According to the performance characteristics in this assembly environment class, no delay will occur if infeasible assembly steps are elimimated. By assigning a value of infinity to the X-Y table travel time for infeasible assembly steps, the assembly planning problem can be modeled as a traveling salesman problem as follows:

$$\text{MIN} \sum_{(i,j)\in FAS} OB(i,j)*X(i,j) + \sum_{(i,j)\in IFAS} RINF*X(i,j) \qquad (4.7)$$

subject to

$$\sum_{\text{all } i} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \qquad (4.8)$$

$$\sum_{\text{all } j} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \quad (4.9)$$

$$U(i)-U(j)+N*X(i,j) < N-1 \qquad \text{for } i = 2,3,\ldots,N;\ j = 2,3,\ldots,N \quad (4.10)$$

$$X(i,j) \text{ non-negative integer} \qquad \text{for all } (i,j) \quad (4.11)$$

where

(i,j): the assembly step from component i to component j

OB(i,j): the X-Y table travel time required for assembly step (i,j)

$$X(i,j) = \begin{cases} 1, & \text{if assembly step } (i,j) \text{ is included in the assembly sequence.} \\ 0, & \text{otherwise.} \end{cases}$$

N: number of components in the PCB

FAS: the set of feasible assembly steps

IFAS: the set of infeasible assembly steps

RINF: extremely large value

Once the optimal assembly sequence is obtained through solving the above mathematical model, the Jumping Assignment Method is used with respect to the optimal assembly sequence, to find the assignment of feeders in the feeder carrier. This assignment and assembly sequence constitute the optimal assembly plan. The assembly planning problem for this assembly environment class is solved.

(HPO=A/N≤NF/SPT=RT)

According to the performance characteristics of this assembly environment class, it is known that zero board delay can not be guaranteed. In order to obtain the optimal assembly plan, two phases of modeling efforts may be required. Parallel to the priority order of the two objectives in this assembly environment class, Phase I concentrates on minimizing board delay while Phase II concentrates on minimizing X-Y table travel time by considering only the assembly sequences with board delay equivalent to the minimal board delay obtained in Phase I. A mathematical model for Phase I can be formulated as a traveling salesman problem as follows:

$$\text{MIN} \sum_{(i,j)\in FAS} BD(i,j)*X(i,j) + \sum_{(i,j)\in IFAS} RINF*X(I,J) \qquad (4.12)$$

subject to

$$\sum_{\text{all } i} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \qquad (4.13)$$

$$\sum_{\text{all } j} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \qquad (4.14)$$

$$U(i)-U(j)+N*X(i,j) \leq N-1 \qquad \text{for } i = 2,3,\ldots,N;\ j = 2,3,\ldots,N \qquad (4.15)$$

$$X(i,j) \text{ non-negative integer} \qquad \text{for all } (i,j) \qquad (4.16)$$

where

(i,j): the assembly step from component i to component j

OB(i,j): the X-Y table travel time required for assembly step (i,j)

$$X(i,j)=\begin{cases}1, & \text{if assembly step } (i,j) \text{ is included in the assembly sequence.}\\ 0, & \text{otherwise.}\end{cases}$$

N: number of components in the PCB

FAS: the set of feasible assembly steps

IFAS: the set of infeasible assembly steps

RINF: extremely large value

BD(i,j): the board delay for assembly step (i,j)

$$BD(i,j)=MAX(OB(i,j)-RT,0)$$

The constraints of the mathematical model for Phase II can be formulated exactly the same as the model for Phase I, with the addition of a constraint. This added constraint is:

$$\sum_{(i,j)\in FAS} BD(i,j)*X(i,j) + \sum_{(i,j)\in IFAS} RINF*X(i,j) = TBD \qquad (4.17)$$

where

TBD: the minimal board delay obtained at Phase I.

The objective function is the same as Eq. (4.7). Solving a TSP is very time consuming if the number of cities exceeds a value of approaximately 30. In order to reduce computation time, it is desired to merge the two models into

one without affecting the optimal assembly plan. The composite model can be formulated as a TSP by changing the coefficients in the objective function. The model can be expressed as follows:

$$\text{MIN} \sum_{(i,j)\in \text{FAS}} (BD(i,j)+OB(i,j)/SF1)*X(i,j) + \sum_{(i,j)\in \text{IFAS}} RINF*X(i,j) \qquad (4.18)$$

subject to

$$\sum_{\text{all } i} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \quad (4.19)$$

$$\sum_{\text{all } j} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \quad (4.20)$$

$$U(i)-U(j)+N*X(i,j) \leq N-1 \qquad \text{for } i = 2,3,\ldots,N;\ j = 2,3,\ldots,N \quad (4.21)$$

$$X(i,j) \text{ non-negative integer} \qquad \text{for all } (i,j) \quad (4.22)$$

where

(i,j): the assembly step from component i to component j

OB(i,j): the X-Y table travel time required for assembly step (i,j)

$$X(i,j)=\begin{cases} 1, & \text{if assembly step } (i,j) \text{ is included in the assembly sequence.} \\ 0, & \text{otherwise.} \end{cases}$$

N: number of components in the PCB

FAS: the set of feasible assembly steps

IFAS: the set of infeasible assembly steps

RINF: extremely large value

BD(i,j): the board delay for assembly step (i,j)

BD(i,j)=MAX(OB(i,j)-RT,0)

SF1: the scaling factor which is the summation of the N largest X-Y table travel times of the feasible assembly steps.

With this kind of arrangement on the coefficients of the objective function, the resultant optimal assembly plan is the same as the one obtained by the two-phase method. This can be shown as follows:

The new coefficient of X(i,j) in the objective function is composed of two parts: the integer part and the decimal part. The integer part is the corresponding board delay while the decimal part is a scaled value representing relative length of the corresponding X-Y table travel. The decimal part is scaled such that the summation of the decimal parts of any assembly sequence is less than one. Therefore, the objective value of any assembly sequence is of the form IA.RB where IA is the summation of integer parts of all the assembly steps included in the assembly sequence and RB is the summation of decimal parts of all the assembly steps included in the assembly sequence. Suppose the minimal objective value reached by solving the above model is AAA.BBB; it is obvious that AAA must be equivalent to the minimal integer part of all potential assembly sequences.

Otherwise, AAA.BBB can not be optimal. Since the integer part of the objective value is the total board delay, the resultant assembly sequence will have minimal total board delay.

When two or more assembly sequences have the same minimal total board delay, AAA.BBB is the one with minimal decimal part among the objective values from AAA to AAA+1. Since the decimal part of the objective value represents equally reduced total X-Y table travel time, the resultant assembly sequence is the one with minimal X-Y table travel time among those assembly sequences with the same minimal total board delay. Hence, the optimal assembly sequence obtained is the same as the one obtained by the two phase method.

Once the optimal assembly sequence is obtained through solving the above mathematical model, application of the Jumping Assignment Method with respect to the resulting assembly sequence, will give the assignment of components in the feeders. This assignment and assembly sequence constitute the optimal assembly plan. The assembly planning problem in this assembly environment class is solved.

(HPO=B/N≤NF/SPT=RT)

According to the performance characteristics of this assembly environment class, it is known that some feasible assembly steps may generate board delay. The priority order of the two objectives is to minimize total X-Y table travel first and then to minimize total delay.

The mathematical model can be formulated as a traveling salesman problem by following an approach similar to that used in (HPO=A/N≤NF/SPT=RT). The mathematical model is expressed as follows:

$$\text{MIN} \sum_{(i,j)\in \text{FAS}} (OB(i,j)+BD(i,j)/SF2)*X(i,j) + \sum_{(i,j)\in \text{IFAS}} RINF*X(i,j) \qquad (4.23)$$

subject to

$$\sum_{\text{all } i} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \qquad (4.24)$$

$$\sum_{\text{all } j} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \qquad (4.25)$$

$$U(i)-U(j)+N*X(i,j) \leq N-1 \qquad \text{for } i = 2,3,\ldots,N;\ j = 2,3,\ldots,N \qquad (4.26)$$

$$X(i,j) \text{ non-negative integer} \qquad \text{for all } (i,j) \qquad (4.27)$$

where

(i,j): the assembly step from component i to component j

OB(i,j): the X-Y table travel time required for assembly step (i,j)

$$X(i,j)=\begin{cases}1, & \text{if assembly step } (i,j) \text{ is included in the assembly sequence.}\\ 0, & \text{otherwise.}\end{cases}$$

N: number of components in the PCB

FAS: the set of feasible assembly steps

IFAS: the set of infeasible assembly steps

RINF: extremely large value

BD(i,j): the board delay for assembly step (i,j)

$$BD(i,j)=MAX(OB(i,j)-RT,0)$$

SF2: the scaling factor which is the summation of the N largest board delays of the feasible assembly steps.

With this kind of arrangement on the coefficients of the objective function, the resultant optimal assembly sequence is the one with minimal total delay under the condition that its total X-Y table travel is minimized first. The proof is the same as that in the previous section.

(HPO=A/N>NF/SPT=BE)

According to the performance characteristics of this assembly environment class, no board delay will occur, and the minimal pick delay is zero. However, assignment of components to the feeders in the feeder carrier must be considered by the model. The assembly planning problem in this assembly environment class can be solved by

concentrating on the second objective--minimizing total X-Y table travel time, with minimal total delay equal to zero as an extra constaint. The basic structure of this model contains three modules. These are:

1. assembly sequence module;
2. assignment module;
3. connection module.

Assembly Sequence Module

This module establishes feasible assembly sequences and can be formulated the same as the constraints of a traveling salesman problem as follows:

$$\sum_{\text{all } i} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \quad (4.28)$$

$$\sum_{\text{all } j} X(i,j) = 1 \qquad \text{for } j = 1,2,\ldots,N \quad (4.29)$$

$$U(i)-U(j)+N*X(i,j) \leq N-1 \qquad \text{for } i = 2,3,\ldots,N;\ j = 2,3,\ldots,N \quad (4.30)$$

$$X(i,j) \text{ non-negative integer} \qquad \text{for all } (i,j) \quad (4.31)$$

where

(i,j): the assembly step from component i to component j

$$X(i,j) = \begin{cases} 1, & \text{if assembly step } (i,j) \text{ is included in the assembly sequence.} \\ 0, & \text{otherwise.} \end{cases}$$

N: number of components in the PCB

Assignment Module

In this assembly environment class, the number of components is greater than the number of feeders in the feeder carrier. The Jumping Assignmnet Method will not guarantee zero pick delay. Therefore, the mathematical model must be formulated to consider the assignment of components to the feeders. It is assumed that each component type can be assigned to one and only one feeder in the feeder carrier. In order to simplify notations, it is assumed that components numbered from 1 to NTT (total number of component types) represent component types 1 to NTT, respectively.

The function of this assignment module is to establish the feasible solution space for the various kinds of assignments. This module can be formulated as follows:

$$\sum_{k=1}^{NTT} W(i,k) = 1 \qquad \text{for } i = 1,2,\ldots,NTT \quad (4.32)$$

$$\sum_{i=1}^{NTT} W(i,k) = 1 \qquad \text{for } k = 1,2,\ldots,NTT \quad (4.33)$$

$$P(i) = \sum_{k=1}^{NTT} k*W(i,k) \qquad \text{for } i = 1,2,\ldots,NTT \quad (4.34)$$

$$P(NTT+C) = P(T(NTT+C)) \qquad \text{for } C=1,2,\ldots,N-NTT \quad (4.35)$$

$$W(i,k) \text{ integer} \quad (4.36)$$

where

NTT: the total number of component types in the PCB.

P(i): the position number of the feeder to which the ith component is assigned.

T(NTT+C): the component type to which the (NTT+C)th component belongs.

$$W(i,k)=\begin{cases}1, & \text{if the ith component is assigned to the Kth feeder}\\ 0, & \text{otherwise}\end{cases}$$

Constraints (4.32), (4.33) and (4.36) force each component type from 1 to NTT to be assigned to one and only one feeder in the feeder carrier. Constraint (4.34) is used to record the positions of feeders to which the various type of components are assigned. The positions of the feeders will be used to calculate the delivery time of a component in the connection module. The function of constraint (4.35) is to assign components of the same type to the same feeder.

An example is presened to facilitate understanding of the structure of the assignment module. Suppose there are 8 components in the PCB and the 8 components belong to 4 different component types. Namely, component 1 is of type 1; component 2 is of type 2; component 3 is of type 3; component 4 is of type 4; component 5 is of type 3; component 6 is of type 1; component 7 is of type 4, and component 8 is of type 3. The corresponding assignment module can be formulated as:

W(1,1)+W(1,2)+W(1,3)+W(1,4)=1

W(2,1)+W(2,2)+W(2,3)+W(2,4)=1

W(3,1)+W(3,2)+W(3,3)+W(3,4)=1

W(4,1)+w(4,2)+W(4,3)+W(4,4)=1

W(1,1)+W(2,1)+W(3,1)+W(4,1)=1

W(1,2)+W(2,2)+W(3,2)+W(4,2)=1

W(1,3)+W(2,3)+W(3,3)+W(4,3)=1

W(1,4)+W(2,4)+W(3,4)+W(4,4)=1

P(1)=1*W(1,1)+2*W(1,2)+3*W(1,3)+4*W(1,4)

P(2)=1*W(2,1)+2*W(2,2)+3*W(2,3)+4*W(2,4)

P(3)=1*W(3,1)+2*W(3,2)+3*W(3,3)+4*W(3,4)

P(4)=1*W(4,1)+2*W(4,2)+3*W(4,3)+4*W(4,4)

P(5)=P(3)

P(6)=P(1)

P(7)=P(4)

P(8)=P(3)

Connection Module

The function of the connection module is to integrate the assembly sequence and assignment modules together to generate solutions for the objective function. The module can be formulated as follows:

$$\text{MIN} \sum_{\text{all } (i,j)} OB(i,j)*X(i,j) \qquad (4.37)$$

subject to

$$RT \geq UT*|P(i)-P(j)| \quad \text{for all } (i,j) \text{ if } X(i,j)=1 \qquad (4.38)$$

where

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

OB(i,j): the X-Y table travel time required for assembly step (i,j)

P(i): the position number of the feeder to which the ith component is assigned.

$$X(i,j)=\begin{cases} 1, & \text{if assembly step } (i,j) \text{ is included in the assembly sequence.} \\ 0, & \text{otherwise.} \end{cases}$$

The right hand side of constraint (4.38) represents the delivery time required to move the feeder of the jth component to the pick point while originally the feeder of the ith component is located at the pick point. The left hand side of this constraint is the normal robot round trip time which must be greater than or equal to the required component delivery time such that pick delay can not occur.

Constraint (4.38) is not linear, but the linearization process can be accomplished by the following constraints:

$$RT*X(i,j)-M*(X(i,j)-1) \geq UT*(P(i)-P(j)) \text{ for all } (i,j) \qquad (4.39)$$

$$RT*X(i,j)-M*(X(I,j)-1) \geq UT*(P(j)-P(i)) \text{ for all } (i,j) \qquad (4.40)$$

where

M: The travel time between two extreme feeders in the feeder carrier.

P(i): the position number of the feeder to which the ith component is assigned.

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

To show that the two constraints (4.39) and (4.40) when used together are equivalent to the nonlinear constraint (4.38), a lemma is introduced and proved first.

Lemma 1: The feasible solution space of a non-linear constraint $A \geq |X-Y|$ is equivalent to the feasible solution of two linear constraints $A \geq X-Y$ and $A \geq Y-X$.

In order to show that the above lemma is true, it is necessary to prove that the non-linear constraint and the two linear constraints represent the same solution space if they are in the same situation.

Situation 1: $X \geq Y$

If $X \geq Y$, the non-linear constraint is actually $A \geq X-Y$ which is the first of the two linear constraints. The second linear constraint $A \geq Y-X$ has a non-positive right hand side and becomes redundant. Thus, at situation 1, the non-linear constraint is equivalent to the two linear constraints.

Situation 2: $Y > X$

If $Y > X$, the non-linear constraint is actually $A \geq Y-X$ which is the second of the linear two constraints. The first linear constraint $A \geq X-Y$ now has negative right

hand side and becomes redundant. Thus, at situation 2, the nonlinear constraint is equivalent to the two linear constraints.

The proof of this lemma is completed.

With the help of Lemma 1, it is clear that constraints (4.39) and (4.40) can be used to replace the following nonlinear constraint:

$$RT*X(i,j)-M(X(i,j)-1) \geq UT*|P(i)-P(j)| \qquad (4.41)$$

Next, it is necessary to show that this constraint is equivalent to constraint (4.38). When X(i,j)=0, constraint (4.41) becomes $M \geq UT*|P(i)-P(j)|$. This constraint is redundant since M is always greater than or equal to $UT*|P(i)-P(j)|$ for any (i,j) by definition. When X(i,j)=1, constraint (4.41) becomes $RT \geq UT*|P(i)-P(j)|$ which is the same as constraint (4.38). We can conclude that the connection module can be linearized as follows:

$$\text{MIN} \sum_{\text{all } (i,j)} OB(i,j)*X(i,j)$$

subject to

$$RT*X(i,j)-M*(X(i,j)-1) \geq UT*(P(i)-P(j)) \quad \text{for all } (i,j)$$

$$RT*X(i,j)-M*(X(I,j)-1) \geq UT*(P(j)-P(i)) \quad \text{for all } (i,j)$$

where

M: The travel time between two extreme feeders in the feeder carrier.

P(i): the position number of the feeder to which the ith component is assigned.

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

By solving the three modules, an optimal assembly plan for this assembly environment class can be reached.

(HPO=A/N>NF/SPT=AC)

According to the performance characteristics of this assembly environment class, no board delay will occur if feasible assembly steps are excluded from the assembly sequence. However, it is not sure whether zero pick delay can be reached. Therefore, the mathematical model must accomodate the coordination of the robot movements, the X-Y table movement and the movement of the feeder carrier. The assembly planning problem can be solved in two phases in parallel to the priority order of the two objectives. That is, the mathematical model in phase I determines the minimal total delay while phase II determines the optimal assembly plan which is the one with minimal X-Y table travel time among those assembly plans with the same minimal total delay identified in Phase I.

The mathematical model of Phase I also has three modules as that of (HPO=A/N>NF/SPT=BE). The assembly

sequence module and assignment module are exactly the same as described in (HPO=A/N>NF/SPT=BE). However, the structure of the connection module is different due to the fact that total delay may not always be zero.

This connection module can be formulated as:

$$\text{MIN} \sum_{(i,j)\in \text{FAS}} PD(i,j) \tag{4.42}$$

subject to

$$PD(i,j)+RT*X(i,j)-M(X(i,j)-1)\geq UT*(P(i)-P(j)) \quad \text{for all } (i,j)\in \text{FAS} \tag{4.43}$$

$$PD(i,j)+RT*X(i,j)-M(X(i,j)-1)\geq UT*(P(i)-P(j)) \quad \text{for all } (i,j)\in \text{FAS} \tag{4.44}$$

$$PD(i,j)>0 \quad \text{for all } (i,j) \in \text{FAS} \tag{4.45}$$

$$\sum_{(i,j)\in \text{IFAS}} X(i,j)=0 \tag{4.46}$$

where

PD(i,j): pick delay of assembly step (i,j)

IFAS: the set of infeasible assembly steps

FAS: the set of feasible assembly steps

P(i): the position number of the feeder to which the ith component is assigned.

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

Constraints (4.43), (4.44) and (4.45) can force PD(i,j) to take a value which is greater than or equal to Max(UT*|P(i)-P(j)|-RT,0) if X(i,j)=1. If X(i,j)=0,

constraint (4.43) and (4.44) become redundant and PD(i,j) can take a non-negative value. Constraint (4.46) actually eliminates infeasible assembly steps from the assembly sequence. Minimizing the summation of all PD(i,j) yields the minimal total pick delay.

Mathematical model for Phase II is of the same structure as that of Phase I. The assembly sequence module and assignment module are not changed while the connection module is formulated as below:

$$\text{MIN} \sum_{(i,j)\in \text{FAS}} OB(i,j)*X(i,j) + \sum_{(i,j)\in \text{IFAS}} RINF*X(i,j) \qquad (4.47)$$

subject to

$$PD(i,j)+RT*X(i,j)-M(X(i,j)-1) \geq UT*(P(i)-P(j)) \quad \text{for } (i,j)\in \text{FAS} \qquad (4.48)$$

$$PD(i,j)+RT*X(i,j)-M(X(i,j)-1) \geq UT*(P(j)-P(i)) \quad \text{for } (i,j)\in \text{IFAS} \qquad (4.49)$$

$$\sum_{(i,j)\in \text{FAS}} PD(i,j)=TPD \qquad (4.50)$$

$$PD(i,j) \geq 0 \qquad (4.51)$$

where

TPD: the total pick delay obtained in Phase I

PD(i,j): pick delay of assembly step (i,j)

IFAS: the set of infeasible assembly steps

FAS: the set of feasible assembly steps

P(i): the position number of the feeder to which the ith component is assigned.

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

(HPO=A/N>NF/SPT=RT)

According to the performance characteristics of this assembly environment class, some feasible assembly steps may generate board delay. Moreover, pick delay may occur. Hence, the connection module should consider potential board delay as well as pick delay. Parallel to the priority order of the two objectives, the model of Phase I is constructed to determine minimal total delay, and the model of Phase II is constructed to determine minimal total X-Y table travel time. The two models contain the same assembly sequence module and assignment module as described in the previous section.

The objective function of the Connection Module can be viewed as:

$$\text{MIN} \sum_{(i,j) \in \text{FAS}} \text{PDI}(i,j)+\text{BDI}(i,j) \qquad (4.52)$$

where

BDI(i,j): The board delay of assembly step (i,j) under the influence of pick delay occurring in the assembly sequence

PDI(i,j): The pick delay of assembly step (i,j) under the influence of board delay occurring in the assembly sequence

The constraints must be established by taking consideration of relationship between the two types of delay. The pick delay at the fixed pick point may be affected by the previous board delay occurring at the place point while the board delay at the place point may be affected by the previous pick delay occurred at the pick point. This will result in a circular chain reaction. Hence, the constraints can be formulated as follows:

$$\sum_{(i,j)\in IFAS} X(i,j) = 0 \qquad (4.53)$$

$$PDI(i,j) \geq 0 \qquad (4.54)$$

$$BDI(i,j) \geq 0 \qquad (4.55)$$

$$RT+PDI(i,j)+BDI(i,j) \geq OB(i,j) \quad \text{for } (i,j)\in FAS \text{ with } X(i,j)=1 \qquad (4.56)$$

$$RT+\sum_{(k,i)\in FAS} BDI(K,i) + PDI(i,j) \geq UT*|P(i)-P(j)| \quad \text{for } (i,j) \text{ FAS with } X(i,j)=1 \qquad (4.57)$$

where

BDI(i,j): The board delay of assembly step (i,j) under the influence of pick delay occurring in the assembly sequence

PDI(i,j): The pick delay of assembly step (i,j) under the influence of board delay occurring in the assembly sequence

IFAS: the set of infeasible assembly steps

FAS: the set of feasible assembly steps

P(i): the position number of the feeder to which the ith component is assigned.

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

Constraint (4.56) states that at fixed place point the summation of the robot normal trip time, the previous pick delay occurred at the pick point and the current board delay must be greater than or equal to the corresponding X-Y table travel time such that the robot can place a component on the PCB. Constraint (4.57) states that at the pick point, the summation of the robot normal round trip time, the previous board delay that occurred at the place point and the current pick delay must be greater than or equal to the delivery time such that the component can be picked up by the robot. Constraint (4.53) eliminates infeasible assembly steps from the assembly sequence. Constraints (4.54) and (4.55) require board delay and pick delay to be nonnegative.

The constraints are not all linear; the linearization process can be accomplished by the following constraints:

$$RT*X(i,j)+PDI(i,j)+BDI(i,j)-M(X(i,j)-1)\geq OB(i,j) \quad \text{for } (i,j)\in FAS \qquad (4.58)$$

$$RT*X(i,j)+\left[\sum_{(k,i)\in FAS} BDI(K,i)\right]+PDI(i,j)-M(X(i,j)-1)\geq UT(P(i)-P(j)) \quad \text{for } (i,j)\in FAS \qquad (4.59)$$

$$RT*X(i,j)+\left[\sum_{(k,i)\in FAS} BDI(K,i)\right]+PDI(i,j)-M(X(i,j)-1)\geq UT(P(j)-P(i)) \quad \text{for } (i,j)\in FAS \qquad (4.60)$$

$$PDI(i,j)\geq 0 \quad \text{for } (i,j)\in FAS \qquad (4.61)$$

$$BDI(i,j)\geq 0 \quad \text{for } (i,j)\in FAS \qquad (4.62)$$

$$\sum_{(i,j)\in IFAS} X(i,j)=0 \qquad (4.63)$$

where

BDI(i,j): The board delay of assembly step (i,j) under the influence of pick delay occurring in the assembly sequence

PDI(i,j): The pick delay of assembly step (i,j) under the influence of board delay occurring in the assembly sequence

IFAS: the set of infeasible assembly steps

FAS: the set of feasible assembly steps

P(i): the position number of the feeder to which the ith component is assigned.

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

Consequently, the connection module in Phase II may be formulated as follows:

$$\text{MIN} \quad \sum_{(i,j)\in FAS} OB(i,j)*X(i,j)+\sum_{(i,j)\in IFAS} RINF*X(i,j) \qquad (4.64)$$

subject to

$$RT*X(i,j)+PDI(i,j)+BDI(i,j)-M(X(i,j)-1)\geq OB(i,j) \quad \text{for } (i,j)\in FAS \tag{4.65}$$

$$RT*X(i,j)+\left[\sum_{(k,i)\in FAS} BDI(K,i)\right]+PDI(i,j)-M(X(i,j)-1)\geq UT(P(i)-P(j)) \quad \text{for } (i,j)\in FAS \tag{4.66}$$

$$RT*X(i,j)+\left[\sum_{(k,i)\in FAS} BDI(K,i)\right]+PDI(i,j)-M(X(i,j)-1)\geq UT(P(j)-P(i)) \quad \text{for } (i,j)\in FAS \tag{4.67}$$

$$PDI(i,j)\geq 0 \quad \text{for } (i,j)\in FAS \tag{4.68}$$

$$BDI(i,j)\geq 0 \quad \text{for } (i,j)\in FAS \tag{4.69}$$

where

BDI(i,j): The board delay of assembly step (i,j) under the influence of pick delay occurring in the assembly sequence

PDI(i,j): The pick delay of assembly step (i,j) under the influence of board delay occurring in the assembly sequence

IFAS: the set of infeasible assembly steps

FAS: the set of feasible assembly steps

P(i): the position number of the feeder to which the ith component is assigned.

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

(HPO=B/N>NF/SPT=BE)

According to the performance characteristics of this assembly environment class, no board delay may occur. The priority order of the two objectives is to minimize total X-Y table travel first and then to minimize total delay. Since the total X-Y table travel time is determined solely by the assembly sequence selected and has nothing to do with the assignment of components to the feeders, the model of Phase I in this case is exactly the same as that in (HPO=AB/N≤NF/SPT=BE).

The assembly sequence module, assignment module and connection module are included in the model of Phase II. The assembly sequence module and assignment module maintain the same structure as described in the previous sections.

The connection module can be constructed as follows:

$$\text{MIN} \sum_{\text{all } (i,j)} PD(i,j)*X(i,j) \qquad (4.70)$$

subject to

$$RT*X(i,j)-M*(X(i,j)-1)\geq UT*(P(i)-P(j)) \quad \text{for all } (i,j) \qquad (4.71)$$

$$RT*X(i,j)-M*(X(I,j)-1)\geq UT*(P(j)-P(i)) \quad \text{for all } (i,j) \qquad (4.72)$$

$$PD(i,j)\geq 0 \qquad (4.73)$$

$$\sum_{\text{all } (i,j)} OB(i,j)*X(i,j)=TOB \qquad (4.74)$$

where

M: The travel time between two extreme feeders in the feeder carrier.

IFAS: the set of infeasible assembly steps

FAS: the set of feasible assembly steps

P(i): the position number of the feeder to which the ith component is assigned.

PD(i,j): pick delay of assembly step (i,j)

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

TOB: is the total X-Y table travel time obtained in Phase I.

(HPO=B/N>NF/SPT=AC)

Based on the same reason as explained in the previous section, the mathematical model of Phase I in this assembly environment class can be formulated as that of (HPO=AB/N≤NF/SPT=AC). The three modules are also required in Phase II with the same assembly sequence module and assignment module. The connection module can be constructed as follows:

$$\text{MIN} \sum_{(i,j)\in \text{FAS}} PD(i,j) \qquad (4.75)$$

subject to

$$PD(i,j)+RT*X(i,j)-M(X(i,j)-1)\geq UT*(P(i)-P(j)) \quad \text{for all } (i,j)\in \text{FAS} \qquad (4.76)$$

$$PD(i,j)+RT*X(i,j)-M(X(i,j)-1)\geq UT*(P(j)-P(i)) \quad \text{for } (i,j)\in FAS \quad (4.77)$$

$$PD(i,j)\geq 0 \quad \text{for } (i,j)\in FAS \quad (4.78)$$

$$\sum_{(i,j)\in IFAS} X(i,j)=0 \quad (4.79)$$

$$\sum_{(i,j)\in FAS} OB(i,j)*X(i,j)=TOB \quad (4.80)$$

where

PD(i,j): pick delay of assembly step (i,j)

IFAS: the set of infeasible assembly steps

FAS: the set of feasible assembly steps

P(i): the position number of the feeder to which the ith component is assigned.

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

(HPO=B/N>NF/SPT=RT)

According to the performance characteristics of this assembly environment class, it is known that some feasible assembly steps may generate board delay, and pick delay may occur. Therefore, the mathematical models should consider both types of delay. However, in Phase I, minimal total X-Y table travel can be determined without considering those two types of delay. The model of Phase I can be formulated exactly the same as that in (HPO=B/N>NF/SPT=AC) if there are

some infeasible assembly steps. Otherwise, the model of Phase I can be formulated as that in (HPO=B/N>NF/SPT=BE).

As usual, there are three modules required in Phase II in which the assembly sequence module and assignment module maintain the same structure. The connection module can be formulated as follows:

$$\text{MIN} \sum_{(i,j)\in FAS} PDI(i,j)+BDI(i,j) \tag{4.81}$$

subject to

$$RT*X(i,j)+PDI(i,j)+BDI(i,j)-M(X(i,j)-1)\geq OB(i,j) \quad \text{for } (i,j)\in FAS \tag{4.82}$$

$$RT*X(i,j)+\left[\sum_{(k,i)\in FAS} BDI(K,i)\right]+PDI(i,j)-M(X(i,j)-1)\geq UT(P(i)-P(j)) \quad \text{for } (i,j)\in FAS \tag{4.83}$$

$$RT*X(i,j)+\left[\sum_{(k,i)\in FAS} BDI(K,i)\right]+PDI(i,j)-M(X(i,j)-1)\geq UT(P(j)-P(i)) \quad \text{for } (i,j)\in FAS \tag{4.84}$$

$$PDI(i,j)\geq 0 \quad \text{for } (i,j)\in FAS \tag{4.85}$$

$$BDI(i,j)\geq 0 \quad \text{for } (i,j)\in FAS \tag{4.86}$$

$$\sum_{(i,j)\in IFAS} X(i,j)=0 \tag{4.87}$$

$$\sum_{(i,j)\in FAS} OB(i,j)*X(i,j)=TOB \tag{4.88}$$

where

BDI(i,j): The board delay of assembly step (i,j) under the influence of pick delay occurring in the assembly sequence

PDI(i,j): The pick delay of assembly step (i,j) under the influence of board delay occurring in the assembly sequence

IFAS: the set of infeasible assembly steps

FAS: the set of feasible assembly steps

P(i): the position number of the feeder to which the ith component is assigned.

RT: normal robot round trip time

UT: unit delivery time of the feeder carrier

CHAPTER V

THE INTEGRATED HEURISTIC METHOD

Due to the complexity of the mathematical models for the assembly planning problem and the large scale nature (in terms of the number of components in the PCBs), it is unrealistic to solve for the true optimal solution. Consequently, a fast heuristic method should be used to obtain reasonably good near-optimal assembly plans. An integrated heuristic method consisting of 5 stages is developed to obtain near-optimal assembly plans for the robotized assembly center. For assembly environment classes (*/N≤NF/*) in which the number of feeders of the feeder carrier is greater than or equal to the number of components in the PCB, only Stage I is needed to obtain the near-optimal assembly plan. For assembly environment classes (HPO=B/N>NF/SPT=BE), (HPO=B/N>NF/SPT=RT) and (HPO=B/N>NF/SPT=AC), Stage I and Stage II are used to obtain the near-optimal assembly plan.

For assembly environment classes (HPO=A/N>NF/SPT=BE), (HPO=A/N>NF/SPT=AC) and (HPO=A/N>NF/SPT=RT), Stage I and Stage II must be used. If the result of Stage II has a zero objective function value, then no other stages are needed. Otherwise, Stage IV must be used in order to reach a

satisfactory assembly plan while additional use of Stages III and V may improve the quality of the assembly plan. The decision on whether or not Stage III and/or Stage V should be used is a user option. The applications of the stages of this integrated heuristic method in different assembly environment classes are summarized in Table 3.

The function of each stage of this integrated heuristic method can be summarized as follows:

Stage I: Find the near-optimal assembly plan for assembly environment classes (*/N≤NF/*). Find the assembly sequence with minimal X-Y table travel for assembly environment classes (*/N>NF/*).

Stage II: Fix the assembly sequence obtained at Stage I and find its corresponding best assignment.

Stage III: Fix the assembly sequence obtained at Stage I and find another K best assignments.

Stage IV: For each assignment obtained at Stage II and III, find its corresponding best assembly sequence. Each pair of assignment and assembly sequence constitute a candidate of satisfactory assembly plan. Choose the best one as the satisfactory assembly plan.

Stage V: Find the near-optimal assembly plan by seeking improvement on the satisfactory assembly plan.

TABLE 3

Application of Integrated Heuristic Method

Assembly Environment Class	Integrated Heuristic Method Stage I	Stage II	Stage III	Stage IV	Stage V
(HPO=AB/N≤NF/SPT=BE)	must	____	____	____	____
(HPO=AB/N≤NF/SPT=AC)	must	____	____	____	____
(HPO=A/N≤NF/SPT=RT)	must	____	____	____	____
(HPO=B/N≤NF/SPT=RT)	must	____	____	____	____
(HPO=A/N>NF/SPT=BE)	must	must	U	R	U
(HPO=A/N>NF/SPT=AC)	must	must	U	R	U
(HPO=A/N>NF/SPT=RT)	must	must	U	R	U
(HPO=B/N>NF/SPT=BE)	must	must	____	____	____
(HPO=B/N>NF/SPT=AC)	must	must	____	____	____
(HPO=B/N>NF/SPT=RT)	must	must	____	____	____

where

U: the use of the stage is determined by user

R: the use of the stage depends on the result of Stage II

The logical procedure of the integrated heuristic method is described in detail in the following sections.

Stage I

Stage I is an integrated heuristic process which seeks the near-optimal solutions for traveling salesman problem with different formulation of the cost matrix. The formulation of the cost matrix depends on the assembly environment class of interest.

Various heuristic methods have been developed in the literature to obtain a near-optimal solution for traveling salesman problem. Stage I supports eight different composite heuristic procedures (CHPs). The user may select one of the eight heuristic procedures, as prefered, to obtain a near-optimal assembly plan.

These eight composite heuristic procedures are (19):

1. Nearest neighbor method starting at all different nodes (CHP1);
2. Farthest insertion method starting at all different nodes (CHP2);
3. Nearest neighbor method (all nodes) first, then 2-opt method (CHP3);
4. Farthest insertion method (all nodes) first, then 2-opt method (CHP4);

5. Nearest neighbor method (all nodes) first, then 3-opt method (CHP5);
6. Farthest insertion method (all nodes) first, then 3-opt method (CHP6);
7. Nearest neighbor method (all nodes) first, 2-opt method second, then 3-opt method (CHP7);
8. Farthest insertion method (all nodes) first, 2-opt method second, then 3-opt method (CHP8).

From Table 1, it is known that Stage I is required for all ten assembly environment classes. The following subsections illustrate how the cost matrix is formulated for each assembly class.

(*/N≤NF/*)

From the analysis of the assembly planning problem, it is known that when the number of components in a PCB is less than or equal to the number of feeders in the feeder carrier, the Jumping Assignment Method can be applied to guarantee zero pick delay. Therefore, the assembly planning problem needs only to take care of the robot movements and the X-Y table travel, and the problem is reduced to a traveling salesman problem.

For assembly environment class (HPO=AB/N≤NF/SPT=BE), the cost matrix is simply the X-Y table travel time matrix

since the assembly sequence with minimal X-Y table travel time will always have zero pick delay if Jumping Assignment Method is applied.

For assembly environment class (HPO=AB/N≤NF/SPT=AC), the cost matrix is a modified X-Y table travel time matrix in which the assembly steps with X-Y table travel time greater than AC are replaced by extremely large values to prevent including these infeasible assembly steps in the assembly sequence.

The priority of the two objectives in assembly environment class (HPO=A/N≤NF/SPT=RT) is minimizing total delay first and then minimizing total X-Y table travel time. In this assembly environment class, some assembly steps may result in board delay. From the mathematical model for this assembly environment class, the cost matrix can be formulated as follows:

1. Form X-Y table travel time matrix XYM(i,j)
2. Form feasible X-Y table travel time matrix FXYM(i,j) by replacing the infeasible assembly steps with extremely large values in the X-Y table travel time matrix.
3. Form board delay matrix BDM(i,j).
4. Form feasible board delay matrix FBDM(i,j).

5. Calculate scaling factor(SF1) which is the summation of the N largest values in FXYM(i,j)
6. Form cost matrix CM(i,j) as follows:

$$CM(i,j)=FBDM(i,j)+FXYM(i,j)/SF1$$

The priority order of the two objectives in (HPO=B/N≤NF/SPT=RT) is: minimizing total X-Y table travel time first and then minimizing total delay. The cost matrix can be formulated the same way as the one for (HPO=A/N≤NF/SPT=RT) except for step 5 and stpe 6. Here step 5 and step 6 changed to:

5. Calculate scaling factor(SF2) which is the summation of the N largest values in FBDM(i,j)
6. Form cost matrix CM(i,j) as follows:

$$CM(i,j)=FXYM(i,j)+FBDM(i,j)/SF2$$

In the assembly environment classes (*/N≤NF/*), once the cost matrix is formulated, Stage I uses one of the eight composite heuristic procedures to find a near-optimal assembly sequence.

Application of the Jumping Assignment Method with respect to the near-optimal assembly sequence will yield the assignment. Hence, near-optimal assembly plan is obtained.

(HPO=A/N>NF/*)

In these assembly environment classes, the assembly problem becomes more complicated because it must take care of the coordination of the robot movements, the X-Y table travel, and also the assignment of components types in the feeders. The mathematical models for this kind of PCB assembly environment classes contain three modules: assembly sequence module, assignment module and connection module. If the three modules are solved simultaneously, the resultant solution is the global optimal plan. However, with the present computer technology and the current solution procedure in integer programming, it is prohibitive to solve the problem as a whole. Instead, it is advantageous to fix solution values for part of the model and then solve for solution values for other parts of the model. The resultant assembly plan is then a local optimum with respect to the fixed part of the model.

It is clear that if assembly sequence is predetermined by some method, then there is at least one pair of local optimal assignments for assigning component types to the feeders. That is, the pair(s) of assignments will generate the same minimal pick delay for the prespecified assembly sequence. Although the priority of these assembly environment classes is to minimize total delay first and

then minimize total delay, the prespecified assembly sequence is still chosen to be the one with minimal X-Y table travel time. The reason is very simple; if the components in the PCB are arranged in such a good order that the components of the same type are grouped nearby, then it is likely that the assembly sequence with the minimal X-Y table travel time will specify many components of the same type being assembled consecutively before assembling another type of component. Consequently, the best assignment for this assembly sequence may have a chance to generate zero pick delay. If the best assignment does generate zero pick delay, this assembly sequence and its corresponding zero pick delay assignment actually constitute the global optimal assembly plan, and no further stages will be needed. However, this ideal condition may not occur; then other stages must be used to find the best assembly plan.

Thus, the formulation of the cost matrix for assembly environment class (HPO=A/N>NF/SPT=BE) is the same as that for assembly environment class (HPO=AB/N≤NF/SPT=BE). The formulation of cost matrix for assembly environment classes (HPO=A/N>NF/SPT=AC) and (HPO=A/N>NF/SPT=RT) is the same as that for assembly environment class (HPO=AB/N≤NF/SPT=AC).

(HPO=B/N>NF/*)

The priority of the objectives is to minimize total X-Y table travel time first and then to minimize total delay. The optimal assembly plans for these assembly environment classes can be obtained easily by following the priority order of the objectives since the determination of the assembly sequence with minimal total X-Y table travel time is not affected by the assignment.

The cost matrix formulation for assembly environment class (HPO=B/N>NF/SPT=BE) is the same as that for (HPO=A/N>NF/SPT=BE). The formulation of cost matrix for assembly environment class (HPO=B/N>NF/SPT=AC) is the same as that for (HPO=A/N>NF/SPT=AC). The formulation of cost matrix for (HPO=B/N>NF/SPT=RT) is the same as that for (HPO=A/N>NF/SPT=RT).

Description of Logical Procedure

Before the logical procedure of Stage I is described, the cost matrices which will be used in the ten assembly environment classes are summarized in Table 4.

The logical procedure of Stage I is described as follows:

Step 1: Read system parameters.

Step 2: Determine assembly environment class.

Step 3: Form the corresponding cost matrix.

TABLE 4

Cost Matrices Used in Stage I

Assembly Environment Class	Cost Matrix Used
(HPO=AB/N≤NF/SPT=BE)	XYM(i,j)
(HPO=AB/N≤NF/SPT=AC)	FXYM(i,j)
(HPO=A/N≤NF/SPT=RT)	FBDM(i,j) + FXYM(i,j) / SF1
(HPO=B/N≤NF/SPT=RT)	FXYM(i,j) + FBDM(i,j) / SF2
(HPO=A/N>NF/SPT=BE)	XYM(i,j)
(HPO=A/N>NF/SPT=AC)	FXYM(i,j)
(HPO=A/N>NF/SPT=RT)	FXYM(i,j)
(HPO=B/N>NF/SPT=BE)	XYM(i,j)
(HPO=B/N>NF/SPT=AC)	FXYM(i,j)
(HPO=B/N>NF/SPT=RT)	FXYM(i,j)

Step 4: Select the composite heuristic procedure.

Step 5: Apply the selected composite heuristic on the cost matrix and obtain an assembly sequence.

Step 6: If the assembly environment class is one of (*/N≤NF/*), then go to Step 7. Else, next stage.

Step 7: Apply Jumping Assignment Method with respect to the assembly sequence obtained in step 5.

Step 8: Record the near-optimal assembly plan which consists of the assembly sequence obtained in step 5 and the assignment obtained in step 7. Calculate corresponding X-Y table travel time and total board delay.

Step 9: Stop.

Stage II

It was shown that the assembly plan for any one of the assembly environment classes (*/N≤NF/*) can be reached at the end of Stage I. Only the remaining assembly environment classes would be put through this stage with an assembly sequence obtained in Stage I.

For assembly environment classes (HPO=B/N>NF/*), the assembly sequence with minimal X-Y table travel time is reached at Stage I. To fulfill the near-optimal assembly plan, it is necessary to find the best assignment for the

assembly sequence at Stage I. The resultant assignment and the assembly sequence obtained at Stage I constitute the near-optimal asssembly plan, and no further stages are required for this assembly environment classes.

For assembly environment classes (HPO=A/N>NF/*), the function of Stage II is to find the best assignment for the assembly sequence with the minimal X-Y table travel time, hoping that zero pick delay assignment can be reached. If zero pick delay assignment results, then the near-optimal assembly plan is reached. Otherwise, further stages are needed.

From the above explanation, it is clear that Stage II is actually an optimal assignment searching process which seeks the assignment of component types to the feeders in the feeder carrier for the assembly sequence obtained at Stage I. Given an assembly step, the X-Y table moves the board between the locations of the starting component and ending component of the assembly step; the feeder carrier moves between the feeders which contain the starting component and the ending component of the assembly step. Once the assembly sequence is predetermined, the only thing of interest concerning the feeder carrier is the number of times the X-Y table travels between each pair of component types instead of the pair of conponent identification number

in the assembly sequence. Since one component type can be assigned to one and only the feeder in the feeder carrier, the optimal assignment searching process can be accomplished by the following integer programming model:

$$\text{MIN} \sum_{(I,J) \text{ with } FREQ(I,J)>0} FREQ(I,J)*PD(I,J) \qquad (5.1)$$

subject to

$$\sum_{I=1}^{NTT} W(I,K) = 1 \qquad \text{for } K=1,2,.....NTT \qquad (5.2)$$

$$\sum_{K=1}^{NTT} W(I,K) = 1 \qquad \text{for } I=1,2,.....NTT \qquad (5.3)$$

$$W(I,K) \text{ integer} \qquad (5.4)$$

$$P(I) = \sum_{J=1}^{NTT} K*W(I,K) \qquad \text{for } I= 1,2,...,NTT \qquad (5.5)$$

$$RT+PD(I,J) \geq UT*(P(I)-P(J)) \qquad \text{for } (I,J) \text{ with } FREQ(I,J)>0 \qquad (5.6)$$

$$RT+PD(I,J) \geq UT*(P(J)-P(I)) \qquad \text{for } (I,J) \text{ with } FREQ(I,J)>0 \qquad (5.7)$$

$$PD(I,J) > 0 \qquad (5.8)$$

where

I,J : are the index number for component types,
I=1,2.....,NTT and J=1,2.....NTT.

(I,J): is the assembly steps from component type I to component type J or from component type J to component type I. J>I is set to uniquely identify the above two types of assembly steps.

FREQ (I,J): the number of times the pair of component types (I,J) is included in the assembly sequence. That is, the number of assembly steps from component type I to component type J plus the number of assembly steps from component type J to component type I.

PD(I,J): the pick delay time for an assembly step (I,J).

$$W(I,K)=\begin{cases}1, & \text{if component type I is assigned to the Kth feeder.}\\ 0, & \text{otherwise.}\end{cases}$$

P(I): the position number of the feeder which is assigned the component type I.

RT: normal robot round trip time.

UT: unit delivery time of the feeder carrier.

The objective function minimizes total pick delay for the predetermined assembly sequence. The pick delay for assembly steps between component type I and component type J is the number of these assembly steps, FREQ(I,J), multiplied by pick delay for individual pick delay required for one assembly step of this kind, PD(I,J).

Constraints (5.2), (5.3) and (5.4) state that each component type can be assigned to one and only one feeder in the feeder carrier as the usual constraints shown in a traditional assignment problem. Constraint (5.5) determines

the position number of the feeder for each component type. Constraints (5.6), (5.7) and (5.8) together determine the pick delay for the assembly step (I,J). Since one component type can not be assigned fractionally nor to more than one feeder, W(I,K) is constrainted to be a 0/1 integer.

The logical procedure of this approach in Stage II can be summarized as follows:

Step 1: Formulate the integer programming model for the assembly sequence obtained at Stage I.

Step 2: Obtain the best assignment and the associated total pick delay for the model.

Step 3: If the assembly environment class is one of (HPO=B/N>NF/*), go to step 5. Else, next step.

Step 4: If total pick delay is zero, go to step 5. Else, go to Stage III or Stage V (The choice is left for the user).

Step 5: The near-optimal assembly plan is reached; this plan consists of the assignment obtained at step 2 and the assembly sequence obtained at Stage I. Calculate corresponding total X-Y table travel time and total board delay.

Step 6: Stop.

The integer programming model of Stage II determines the optimal assignment for the assembly sequence obtained at Stage I. The size of this model increases as the number of components as well as the number of component types increases. At a point this integer programming will become infeasible either because of tremedous computation time required or because the size of the model exceeds the capability of the integer programming computer package. For example, the model requires 225 variables to be defined as 0/1 integers if 15 component types are involved in a PCB. The LINDO computer package can handle only 200 0/1 integer variables. That means a PCB with 15 component types or more may not be run by a LINDO computer package.

If the integer programming model size is prohibitive, the integrated heuristic method can provide a quick searching method for use in Stage II. The quick searching method will be referred to as Quick Stage II from now on while the integer programming approach of Stage II will be referred to as IP Stage II.

Quick Stage II is designed on the basis of the interaction frequency among the components of a given assembly sequence as well as pairwise changes of the feeders of the component types.

To facilitate description and understanding of the design of Quick Stage II, several terms are defined and introduced as follows:

1. Interaction of two component types: Given an assembly sequence, each assembly step starts from one component and ends with another component. If the two components are of different types, then the two component types are said to have an interaction in the given assembly sequence.
2. Interaction Frequency: The number of times two component types interact in an assembly sequence.
3. Interaction Frequency Matrix (FREQM): The matrix in which cell (i,j) represents the interaction frequency between component type I and type J if J is greater than I. Otherwise, the cell (i,j) is assigned a value of zero.
4. Total Interaction sum of a component type (TSUM(I)): The total number of interactions between component type I and all other component types in an assembly sequence.
5. Chain of assigned component types: Quick Stage II assigns component types to feeders iteratively. Moreover, the component types are assigned to occupy a block of consecutive feeders in the feeder carrier.

Hence, the assigned component types form a chain which is referred to as the chain of assigned component types.

6. Seed component type: The first component type member in the chain of assigned component types. That is, the component type which is first considered to occupy a feeder position. In Quick Stage II, every component type can be the seed component type once.
7. Partial interaction sum of an unassigned component type (PSUM(I)): The total interaction frequency between the unassigned component type I and all the other component types in the chain of assigned component types.
8. 2-Switching total pick delay matrix (PDCM): Given an assignment of component types to the feeders and an assembly sequence, cell (i,j) of the matrix represents total pick delay if component type i is assigned to component type j's feeder while component type j is reassigned to component type i's original feeder.

Generally speaking, lower total pick delay may be obtained by assigning component types with intensive interactions in a given assembly sequence to a nearby block of feeders in the feeder carrier. Once all component types

are assigned to feeders, it is possible to reduce total pick delay by iteratively performing pairwise changes of feeders of the component types.

On the basis of this idea, Quick Stage II is designed to contain three main phases:(1) the selection phase; (2) the assignment phase and (3) the improvement phase.

Given a seed component type, the selection phase iteratively finds a component type among the unassigned component types to join the chain of assigned component types. The component type is selected as the one with the largest partial interaction sum with respect to the current chain of assigned component types. Ties may be broken by choosing the component with the least total interaction sum.

When the selection is made, the assignment phase starts by considering whether the selected component type should join the left end or right end of the chain of assigned component types. If the selected component type has interaction frequency with the component type in the current left end of the chain exceeding that with the component type in the current right end of the chain, then the selected component type is assigned to the left end of the chain. Otherwise, it is assigned to the right end.

The selection and assignment phases continue until all component types are assigned to the chain. Next, the

component type in the left end of the chain is assigned to feeder 1, the other component types are assigned to feeders according to their position in the chain. The component type in the right end of the chain is assigned to the NTTth feeder in the feeder carrier. At the point of time, total pick delay with respect to this assignment and assembly squence is recorded.

Another component type is treated as the seed component type, and the selection phase and assignment phase is reinitiated. Corresponding total pick delay is recorded when these phases are finished.

When all component types have been considered as the seed component, the assignment with the smallest total pick delay is chosen as the preliminary assignment of component types to the feeders. The improvement phase will then start.

A 2-switching total pick delay matrix is formulated with respect to the preliminary assignment just obatined and the assembly sequence obtained at Stage I. The cell (i,j) with the smallest total pick delay in the matrix is then identified. If the corresponding total pick delay is less than that of the preliminary assignment, pairwise interchange of the feeders of component types I and J is executed. The resultant assignment is then treated as the

preliminary assignment. Another 2-switching total pick delay matrix is established with respect to the new preliminary assignment. These pairwise changes are performed iteratively until no improvement can be made. The latest preliminary assignment is then treated as the near-optimal assignment for the assembly sequence obtained at Stage I.

The logical structure procedure of Quick Stage II can be summarized as follows:

Step 1: Form interaction frequency matrix FREQM according to the assembly sequence obtained at Stage I.

Step 2: Calculate total interaction sum for each component type on the basis of FREQM.

Step 3: Let ISED=0.

Step 4: Let ISED=ISED+1.

Step 5: Assign the seed component type ISED to the second position in the chain of the assigned component types. The first position is left empty.

Step 6: Let ICA=1.

Step 7: Let ICA=ICA+1.

Step 8: Calculate partial interaction sum for each of the unassigned component types.

Step 9: Select the component type with the largest partial interaction sum be the next one to join the chain of assigned component types. Break ties by choosing the one with the smallest total interaction sum.

Step 10: If ICA=2, assign the selected component type to the third position in the chain and go to Step 7. Else, calculate LEND as the interaction frequency between the selected component type and the component type in the second position of the chain.

Step 11: Calculate REND as the interaction frequency between the selected component type and the component type in the ICAth position of the chain.

Step 12: If LEND > REND, next step. Else, go to step 14.

Step 13: Assign the selected component type to the first position (the left end) of the chain. Move all assigned component types to their corresponding right hand side positions in the chain. Go to step 15.

Step 14: Assign the selected component type to the (ICA+1)th position (the right end) of the chain.

Step 15: If ICA < NTT, go to step 7. Else, next step.

Step 16: Move all component types in the chain to their corresponding left hand side positions in the chain. The position number of a component type is now its feeder position number. Record the assignment.

Step 17: Compute and record total pick delay with respect to the assignment just obtained.

Step 18: If ISED < NTT, go to step 4. Else, next step.

Step 19: Choose the assignment with the smallest total pick delay as the preliminary assignment (IASNP). Record PDIP as the corresponding total pick delay.

Step 20: Established a 2-switching total pick delay matrix (PDCM) with respect to IASNP and FREQM.

Step 21: Identify the cell with the smallest value in the upper triangular portion of the PDCM. Record its row index as IBEG and column index as IEND.

Step 22: If PDCM(IBEG,IEND) < PDIP, next step. Else, go to step 24.

Step 23: Perform pairwise interchange. Assign component type IBEG to component type IEND's feeder position and reassign component type IEND to component type IBEG's original feeder position. Let IASNP be the new assignment. Let PDIP be the corresponding total pick delay. Go to step 20.

Step 24: If the assembly environment class is one of (HPO=B/N>NF/*), go to step 26. Else, next step.

Step 25: If total pick delay is zero, go to step 26. Else, go to Stage III or Stage V (the choice is left for the user).

Step 26: The near optimal assembly plan consists of the the assignment obtained at Step 23 and the assembly sequence obtained at Stage I. Calculate total X-Y table travel time and total delay, and then stop.

Stage III

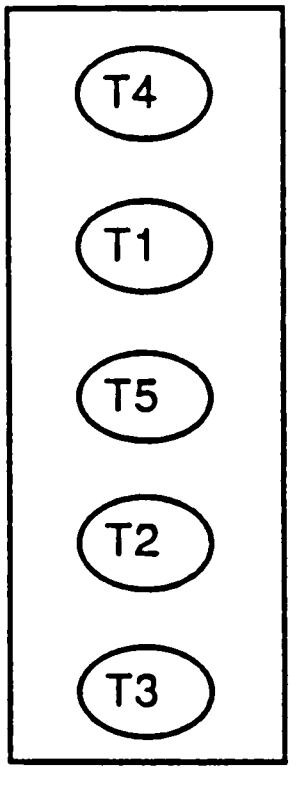
With the use of Stage I and Stage II, a near-optimal assembly plan for (*/N<NF/*) and (HPO=B/N>NF/*) can be reached. Moveover, if Stage II yields zero pick delay assignment for (HPO=A/N>NF/*), the near-optimal assembly plan is also reached. Therefore, Stage III can be used when zero pick delay assignment is not reached for (HPO=A/N>NF/*).

When the assembly sequence with minimal X-Y table travel time can not result in a zero pick delay assignment, it is hoped that the assembly sequence in the final optimal assembly plan may have a reasonably low X-Y table travel time. Thus, the optimal assembly sequence may be the one with many assembly steps in common with the assembly sequence which has the minimal X-Y table travel. By fixing the best assignment with respect to the original assembly sequence obtained at Stage I, a new assembly sequence can be obtained by switching only a few assembly steps which have pick delay in the original assembly sequence with some other assembly steps which are not included in the original assembly sequence. These newly added assembly steps are the ones which will generate the smallest possible pick delay, if not zero pick delay, at this assignment. This edge-switching process is done at Stage IV.

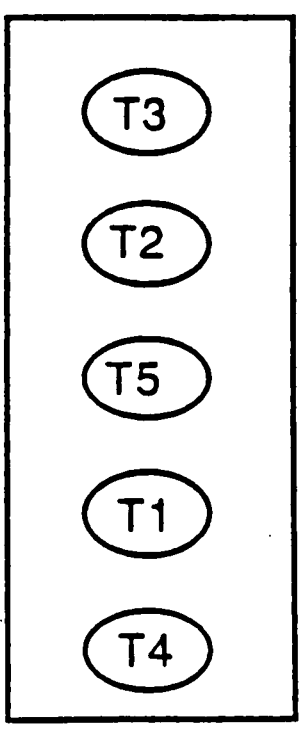
Unfortunately, sometimes the newly added assembly steps may result in comparatively large X-Y table travel in order to avoid pick delay at the assignment of STAGE II. To compensate for this situation, Stage III seeks another K good assignments with respect to the original assembly sequence of Stage I. These assignments provide more opportunity to find a new assembly sequence whose newly added assembly steps have less X-Y table travel.

When a local optimal assignment is reached by IP Stage II, there always is a counter assignment which has the same minimal objective function value. This counter assignment is the reverse assignment of the original optimal assignment. The following example illustrates this relationship. Suppose there are five component types and five component feeders. The original local optimal assignment is to assign component type 1 to feeder 2, 2 to 4, 3 to 5, 4 to 1, and 5 to 3. Then, the counter optimal assignment is 1 to 4, 2 to 2, 3 to 1, 4 to 5 and 5 to 3. The two local optimal assignment are depicted in Figure 14.

If IP Stage II is used, the process of obtaining an additional K best assignments for a predetermined assembly sequence can be done by adding a pair of optimum elimination constraints into the model iteratively when a current local optimum is reached. One constraint is for the current local



(a) Original Optimal Assignment



(b) Counter Optimal Assignment

Figure 14: Paired Local Optimal Assignments

optimal assignment, and one is for its counter assignment. The function of the optimum elimination constraints is to cut off the pair of local optimal assignments from further consideration without affecting other integer solutions in the feasible solution space. The optimum elimination constraint is of the form

$$\sum_{I=1}^{NTT} W(I,P(I\$)) \leq NTT-1$$

where

P(I$): The position number of the feeder which contains component type I determined from the current local optimal assignment.

NTT: The total number of component types in the PCB.

The procedure of IP Stage III can be summarized as follows:

Step 1: User determines the K value.

Step 2: Let ITIMES=2.

Step 3: Identify the reverse assignment of the assignment obtained at Stage II.

Step 4: Add a pair of optimum elimination constraints with respect to the assignment obtained at IP Stage II, into the integer programming model of Stage II. Record resultant assignment.

Step 5: If ITIMES$\geq$K+1, go to next stage. Else, next step.

Step 6: Let ITIMES=ITIMES+1.

Step 7: Obtain the current best assignment from the current integer programming model.

Step 8: Identify the reverse assignment of the assignment obtained at step 7.

Step 9: Add a pair of optimum elimination constraints, with respect to the pair of assignments just obtained, into the current integer programming model. Go to step 5.

Whenever Quick Stage II is used, another K best assignments with respect to the original assembly sequence obtained at stage I are selected as the K assignments associated with the K smallest cells in the latest 2-switching total pick delay matrix of Quick Stage II.

The logical procedure of Quick Stage III can be summarized as follows:

Step 1: User determines the K value.

Step 2: Let ITIMES=2.

Step 3: Identify the cell with the smallest value in the PDCM. Record the row index of this cell as IBEG and record the column index of this cell as IEND.

Step 4: Perform pairwise interchange operation. That is, the feeders of the component types IBEG and IEND are exchanged.

Step 5: If ITIMES$\geq$K+1, go to next stage. Else, next step.

Step 6: ITIMES=ITIMES+1

Step 7: Replace the cell identified at step 3 by RINF which is an extremely large value. Go to Step 3.

Stage IV

When the assignment of the component types to the feeders is predetermined, the way of searching the best assembly sequence for the fixed assignment can be formulated as a traveling saleman problem with respect to a composite cost matrix. This composite cost matrix is composed of two matrices: The X-Y table travel matrix of the feasible assembly steps and the pick delay matrix of the feasible assembly steps. In order to minimize the total delay first and then to minimize total X-Y table travel as requested in (HPO=A/N>NF/*), a scaling factor is needed to merge the two matrices into the composite cost matrix. The scaling factor is the summation of the N largest X-Y table travel times of the feasible assembly steps. Mathematically, the composite cost matrix can be expressed as follows:

CM(i,j)=FPDM(i,j)+FXYM(i,j)/SF1

where

CM(i,j): the cost matrix

FPDM(i,j): the feasible pick delay matrix

FXYM(i,j): the feasible X-Y table travel time matrix

SF1: the summation of the N largest values in X-Y table travel time matrix

Application of any of the eight composite heuristic procedure can obtain a local near-optimal assembly sequence. The logical procedure of Stage IV can be summarized as follows:

Step 1: Read system parameters.

Step 2: Read the (K+1) best assignments obtained at Stage II and Stage III.

Step 3: Let ITIMES=0.

Step 4: Let ITIMES=ITIMES+1.

Step 5: Form the composite cost matrix for the ITIMES assignment.

Step 6: Apply user selected composite heuristic procedure on the composite cost matrix to obtain an assembly sequence.

Step 7: The resultant assembly sequence and the (ITIMES)th assignment constitute a local near-optimal assembly plan. Calculate corresponding total pick delay, total board delay and total X-Y table travel.

Step 8: If ITIMES=K+1, go to next step. Else, go to Step 4.

Step 9: Choose the best assembly plan as the satisfactory assembly plan among the (K+1) local near-optimal assembly plans just obtained. The selection follows the priority order of the two objectives: minimizing total delay first and then minimizing total X-Y table travel.

Step 10: User determines whether Stage V is needed. If it is decided to include Stage V, go to next stage. Otherwise, end.

Stage V

At the end of Stage IV, a satisfactory assembly plan is selected among the K+1 assembly plans. It is obvious that the larger the K value is, the better the satisfactory assembly plan is. However, as the value of K increases, computational effort also increases. Stage V seeks improvements in the assembly plan from Stage IV for smaller K value.

One way of doing this is to selectively screen the assembly sequences from the assembly sequence with lowest X-Y table travel time and locate corresponding local optimal assignment until an assembly plan better than the Stage IV assembly plan is reached or until the upper bound of the screening process is exceeded. If a better-than-satisfactory assembly plan is reached, further improvement may be obtained by fixing its assignment to find its corresponding local optimal assembly sequence.

The selective screening process is achieved by the concept of 2-opt method but with reverse order. The process of locating the best assignment for a selected sequence is

heuristically solved by manipulating a 2-switching total pick delay matrix. This section describes how this is implemented.

Before introducting the logic procedure of this stage, it is necessary to clarify several terms which will be used to describe the logical procedure of this stage.

1. Slant Matrix (SDM): The Slant Matrix is the reconfiguration of the feasible X-Y table travel time matrix. The Slant Matrix for a given sequence is organized such that the ith row corresponds to the ith node in this sequence and the jth column corresponds to the jth node in the sequence. The corresponding X-Y table travel time for each assembly step in the sequence will then become the slant of the matrix.
2. Inversion: An inversion reverses the order of a partial sequence of a given complete assembly sequence. For example, suppose the given assembly sequence is 1,2, i0, il, ...,ik, iK+1..., n, then inversion (il,ik) will invert the given sequence into the new sequence: 1,2,...,i0, ik,..., il, ik+1, ..., n.
3. Inversion cost matrix (IM): The values of the cells in the inversion cost matrix are the incremental X-Y table travel time if an inversion is performed at those cells. For example, IM(il, ik)=SDM(i0, ik)+ SDM(il, ik+1)-SDM(i0, il)-SDM(ik, iK+1).

4. 2-switching total pick delay matrix (PDCM): Given an assignment of component types to the feeders and an assembly sequence, the cell(i,j) of the matrix represents the total pick delay if component type i is assigned to component type j's feeder while component type j is reassigned to component type i's feeder.
5. Standard assembly sequence: The assembly sequence with the lower bound total X-Y table travel time for the screening process. At the beginning, it is the assembly sequence obtained at Stage I.
6. Standard assignment: The best assignment obtained for the standard assembly sequence. That is, the assignment obatined at the end of Stage II.
7. Standard assembly plan: The assembly plan composed of the standard assembly sequence and the standard sequence.
8. Satisfactory assembly plan: The assembly plan obtained at the end of Stage IV.
9. Upper bound of screening process: The total X-Y table travel time of the satisfactory assembly plan.
10. Upper bound of locating process: The total pick delay of the standard assembly plan.
11. Lower bound of locating process: The total pick delay of the satisfactory assembly plan.

Based on the standard assembly sequence, a slant matrix and an inversion cost matrix are estabilished. Each cell in the inversion cost matrix represents the cost increase of a new assembly sequence which inverts the order of a partial sequence whose starting and ending nodes are identified by the row index and column index of the cell, respectively. The screening process screens the assembly sequences in increasing order of the cell values in the inversion cost matrix.

When a new assembly sequence is considered by the screening process, the process of heuristically locating the best assignment for this candidate sequence begins.

For the first candidate assembly sequence, the 2-switching total pick delay matrix is formulated with respect to the standard assembly sequence and the standard assignment. The process then identifies the minimal cell value in this 2-switching total pick delay matrix. If the total pick delay of this cell is no less than the upper bound of the locating process, the candidate sequence is considered not being able to generate an assembly plan better than the given standard assembly plan. The screening process is resumed to find the next candidate assembly sequence which is the next smallest cell in the given inversion cost matrix. On the other hand, if the identified

cell in the 2-switching total delay matrix has a lower value than the upper bound of the locating process, that means an assembly plan better than the given standard assembly sequence is found and the candidate assembly sequence becomes new standard assembly sequence. The locating process continues to find a better assignment for the candidate assembly sequence. This is done by replacing the standard assignment with the newly identified assignment and establishing a new 2-switching total pick delay matrix until a better assignment can not be found. At this point, the resultant total pick delay is compared against the total pick delay of the satisfactory assembly plan. If it is greater than the total pick delay of the satisfactory assembly plan, that means the new standard assembly plan is not better than the given satisfactory assembly plan and the screening process is resumed.

This time, the screening process starts from establishing a new slant matrix with respect to the new standard assembly sequence. A new inversion cost matrix is also established. Any cell with negative inversion cost will be eliminated from the inversion cost matrix to prevent selecting those sequences which had been considered. On the other hand, if the new upper bound is less than or equal to the total pick delay of the satisfactory assembly sequence,

then a better- than-satisfactory assembly plan is reached. Further improvement is then attemped by fixing the standard assignment, which is actually the best assignment of the better-than-satisfactory assembly plan, to find its corresponding optimal assembly sequence. The resultant assembly sequence and the standard assignment constitute the near-optimal assembly plan.

The step-by-step procedure of the backup stage is summarized as follows:

Step 1: Initialization

(1) Treat the assembly sequence of the satisfactory plan obtained at Stage IV as the satisfactory assembly sequence (SEQOR).

(2) Treat the assignment of the satisfactory plan obtained at Stage IV as the satisfactory assignment (IASNOR).

(3) Treat the total X-Y table travel time of the satisfactory plan as the upper bound of the screening process (TLOR).

(4) Treat the total pick delay of the satisfactory plan as the lower bound of the locating process (PDOR).

(5) Treat the assembly sequence obtained at Stage I as the standard assembly sequence (SEQO).

(6) Treat the assignment obtained at Stage II as the standard assignment (IASNO).

(7) Treat the total X-Y table travel time of the standard assembly sequence as the lower bound of the screening process (TLO).

(8) Treat the total delay of the standard assembly plan as the upper bound of locating process.

Step 2: Based on SEQO, form the slant matrix.

Step 3: Based on SEQO and the slant matrix, form inversion cost matrix where all cells with negative values are replaced by RINF which is an extremely large value.

Step 4: Identify the minimal value in the inversion cost matrix. Record its row index and column index as INVT1 and INVT2, respectively.

Step 5: Compute total X-Y table travel time of the new assembly sequence TL1 as TLO plus the value identified at step 4. If TL1 is greater than or equal to TLOR, go to step 16. Else, next step.

Step 6: Form the candidate assembly sequence (SEQ1) by performing inversion on the partial sequence starting at the (INVT1)th node and ending at the (INVT2)th node of SEQO.

Step 7: Based on IASNO and SEQ1, form the 2-switching total pick delay matrix (PDCM(i,j)).

Step 8: Improve the assignment for SEQ1 by continually evaluating and updating PDCM(i,j) until no further

improvement can be made. The resultant assignment is called near optimal assignment (IASN1) for SEQ1. The corresponding total pick delay is named PD1P.

Step 9: If PD1P is greater than or equal to the upper bound of the locating process (PDO), then let IM(INVT1,INVT2) and IM(INVT2,INVT1) be very large values; go to step 4. Else, next step.

Step 10: If PDIP is greater than PDOR, then let SEQO=SEQ1, TLO=TL1, IASN0=IASN1, PDO=PD1P; go to step 2. Else, next step.

Step 11: Form feasible pick delay matrix FPDM(i,j) with respect to IASN1.

Step 12: Calculate the scaling factor (SF1) which is the summation of the N largest values in the feasible X-Y table travel time matrix (FXYM(i,j)).

Step 13: Form the composite cost matrix (CM(i,j)) by the following formula: CM(i,j)=FPDM(i,j)+FXYM(i,j)/SF1.

Step 14: Treat SEQ1 as the initial sequence, apply 2-opt method and 3-opt method on the composite cost matrix to obtain a better sequence. This new sequence is still recorded as SEQ1.

Step 15: Output assembly sequence SEQ1 and assignment IASN1 as the near-optimal assembly plan. Go to step 17.

Step 16: Output the satisfactory assembly plan as the near-optimal assembly plan.

Step 17: End.

The computer programs of this integrated heuristic method are documented in the Department of Industrial Engineering at Texas Tech University.

CHAPTER VI

EVALUATION OF THE INTEGRATED HEURISTIC METHOD

The Experiment

The experiment to obtain the performance of the integrated heuristic method was performed for 35 pseudo boards at six different hypothetical robotized assembly centers. These 35 pseudo boards, of different numbers of components and different numbers of component types, were generated by calling GGDT subroutine from IMSL computer package. The number of components (N) and the number of component types (NTT) for the 35 pseudo boards are summarized in Table 5.

The X coordinates of the boards were generated from a discrete uniform distribution with range from 1 to 350 while the Y coordinates were generated from a discrete uniform distribution with range from 1 to 400. The type identification numbers for the first NTT components in every board were specified to be 1 to NTT, consecutively. The type identifications for the remaining components in the boards were generated from a discrete uniform distribution with range from 1 to NTT. The parameters of the six hypothetical robotized PCB assembly centers are summarized in Table 6.

TABLE 5

Description of Pseudo Boards

Board no.	Number of Components (N)	Number of Component Types (NTT)
1 - 5	10	6
6 - 10	20	6
11 - 15	30	6
16 - 20	40	6
21 - 25	20	9
26 - 30	30	9
31 - 35	40	9

TABLE 6

The Six Hypothetical Assembly Centers

Parameters	Hypothetical Assembly Centers					
	1	2	3	4	5	6
BE (milli-sec.)	930	930	930	930	930	930
AC (milli-sec.)	980	600	980	980	600	980
RT (milli-sec.)	1000	1000	500	1000	1000	500
UT (milli-sec.)	400	400	200	400	200	400
V* (mm/sec.)	540	540	540	540	540	540
PT** (mm/sec.)	100	100	50	100	100	50
NF	40	40	40	9	9	9

* V is the speed of X-Y table
** PT is the time needed for robot to pick/place a component

When the boards are assembled by center 1, the assembly environment is (HPO=AB/N≤NF/SPT=BE). When the boards are assembed by center 2, the assembly environment is (HPO=AB/N≤NF/SPT=AC). When the boards are assembled by center 3, the assembly environment is (HPO=A/N≤NF/SPT=RT) or (HPO=B/N≤NF/SPT=RT). When the boards are assembled by center 4, the assembly environment is (HPO=A/N>NF/SPT=BE) or (HPO=B/N>NF/SPT=BE). When the boards are assembled by center 5, the assembly environment is (HPO=A/N>NF/SPT=AC) or (HPO=B/N>F/SPT=AC). When the boards are assembled by center 6, the assembly environment is (HPO=A/N>NF/SPT=RT) or (HPO=B/N>NF/SPT=RT).

This integrated heuristic method consists of five stages. IP Stage II and IP Stage III were solved by LINDO computer package on a VAX 11/780 computer while the other stages were coded in FORTRAN. Statistics were collected on total X-Y table travel time, total pick delay and total board delay.

The evaluation was conducted in four phases. In phase I, the performance results of the eight composite heuristic procedures of the integrated heuristic method in assembly environment (*/N≤NF/*) were recorded and compared against the best known solutions. A set of Friedman tests was performed to see if there is significant difference among

the heuristic procedures. The strongest composite heuristic procedures were used in phase II.

In phase II, the performance results of the strongest composite heuristic procedures in assembly environment classes (HPO=B/N>NF/*) were recorded and compared against the best known solutions. This evaluation task was performed in three experiments. Experment II.1 recorded performance of the integrated heuristic method with IP Stage II on boards of 6 component types in assembly environment classes (HPO=B/N>NF/*). Experiment II.2 recorded performance of the integrated heuristic method with Quick Stage II on boards of 6 component types in those Assembly environment classes. Experiment II.3 recorded performance of Quick Stage II and IP Stage II on boards of 9 component types. The effectiveness and efficiency of Quick Stage II and IP Stage II were compared.

In phase III, the performance of the integrated heuristic method in assembly environment classes (HPO=A/N>NF/*) were recorded and compared against the best known solutions. Since a large amount of CPU time was needed to obtain a best known solution in these assembly environment classes, only one of the strongest composite heuristic procedures was performed on boards 1 to 15.

As explained in chapter 5, this integrated heuristic method provides flexibility for the user to determine how to use this integrated heuristic method. For example, a user may decide to have 9 best assignments at stage III and not to use stage V while another user may decide not to run stage III but to include stage V. Therefore, it is decided to get statistics of the integrated heuristic method under different uses of the stages. Two experiments were conducted in this phase. Experiment III.1 recorded the performace when stage V was not used. The numbers of assignments used at stage III were 0, 4 and 9. Second, Experiment III.2 examined the performance of the integrated heuristic method when stage V was included.

In phase IV, the integrated heuristic method was applied to an actual printed circuit board with 109 components and 47 component types. The performance was recorded and compared against solution lower bound.

Experiment I: Performance Evaluation in (*/N≤NF/*)

To evaluate the performance of the integrated heuristic method in assembly envoironment classes (*/N≤NF/*), boards 1 to 20 were assembled in the first three hypothetical assmbly centers. As described in the previous chapter, Stage I

alone is required to obtain the near-optimal assembly plans. Stage I was coded in FORTRAN and run on an IBM 3081 computer.

This section presents four sets of computational results. The first set, second set, third set and fourth set evaluate the performance of the eight composite heuristic procedures in assembly environment classes (HPO=AB/N≤NF/SPT=BE), (HPO=AB/N≤NF/SPT=AC), (HPO=A/N≤NF/SPT=RT) and (HPO=B/N≤NF/SPT=RT), respectively.

The best known solutions were obtained on the basis of a modified Eastman algorithm. First, the best solution among the solutions obtained by the eight composite heuristic procedures were treated as the upper bound in this modified Eastman algorithm. If the algorithm reached the optimal solution within 45 CPU seconds, the best known solution was the optimal solution. Otherwise, the run was ended after 45 seconds of CPU time, and the best solution reached at that time was treated as the best known solution. The modified Eastman algorithm is listed in Appendix A.

Computational Results: (HPO=AB/N≤NF/SPT=BE)

According to the performance characteristics in this assembly environment class, it is true that total delay can

always be zero. In Table 7, the performance of the eight different composite heuristic procedures on total X-Y table travel are compared against the best known solutions.

Their average performance on boards of 10, 20, 30 and 40 components are summarized in Table 8. Figure 15 depicts the trend of these average performance. It can be seen from Figure 15 that the average performance of each of the eight composite heuristic procedures declines as the number of components increases. Moreover, the first composite heuristic procedure, the nearest neighbor method, is the worst heuristic procedure in all 4 cases.

The eight composite heuristic procedures are actually based on two different initial tour construction procedures. These are nearest neighbor method and fartherest insertion method. Due to the nature of the four improvement methods, 3-opt method is not worse than 2-opt method when both methods are applied to the same initial tour. Therefore, part of the performance relationship of the eight composite heuristic procedures can be summarized as follows:

$$CHP1 \leq CHP3 \leq CHP7$$

$$CHP1 \leq CHP5$$

$$CHP2 \leq CHP4 \leq CHP8$$

$$CHP2 \leq CHP6$$

where

CHPN: the Nth composite heuristic procedure

TABLE 7

Computational Results: (HPO=AB/N≤NF/SPT=BE)

Board No.	N	BEST Known Solution	Composite Heuristic Procedures 1 (per-cent over)	2 (per-cent over)	3 (per-cent over)	4 (per-cent over)	5 (per-cent over)	6 (per-cent over)	7 (per-cent over)	8 (per-cent over)	Overall CPU (sec.)
1	10	1582	0	0	0	0	0	0	0	0	0.20
2	10	2234	5.86	0	0	0	0	0	0	0	0.21
3	10	2186	3.06	0	0	0	0	0	0	0	0.21
4	10	1996	6.86	0	0	0	0	0	0	0	0.26
5	10	1983	0	0	0	0	0	0	0	0	0.26
6	20	1966	0	0	0	0	0	0	0	0	0.98
7	20	2384	3.64	0	0	0	0	0	0	0	0.86
8	20	2718	1.10	0	0	0	0	0	0	0	0.84
9	20	2558	3.52	0.08	0	0.08	0	0	0	0	0.88
10	20	2618	7.45	1.45	1.72	1.45	1.45	1.45	0	1.45	0.89
11	30	3516	2.90	2.82	0.26	2.82	0.26	2.82	0.26	2.82	3.06
12	30	3354	13.42	1.61	0	0	0	0	0	0	3.30
13	30	3141	1.88	0.86	0	0	0	0	0	0	3.31
14	30	3137	16.16	0.96	5.90	0.96	0	0	0	0	3.28
15	30	3093	5.24	2.42	0.32	2.42	0	0	0	0	3.32
16	40	3765	17.42	3.85	3.64	3.11	0	0.61	2.36	0.61	8.64
17	40	3538	10.34	2.34	1.07	2.34	0	2.22	0	2.22	8.54
18	40	3541	8.13	4.77	1.80	1.61	0.45	0	0.45	0	8.32
19	40	3593	7.27	0.31	0.56	0.31	0.56	0.19	0.56	0.19	7.62
20	40	4016	6.90	2.39	2.48	0.70	0	0.70	0.75	0.70	8.18

TABLE 8

Average Performance: (HPO=AB/N≤NF/SPT=BE)

Number of Components on Board	Composite Heuristic Procedures CHP1 (per-cent over)	CHP2 (per-cent over)	CHP3 (per-cent over)	CHP4 (per-cent over)	CHP5 (per-cent over)	CHP6 (per-cent over)	CHP7 (per-cent over)	CHP8 (per-cent over)
10	3.15	0	0	0	0	0	0	0
20	3.14	0.31	0.34	0.31	0.29	0.29	0	0.29
30	7.42	1.73	1.30	1.56	0.05	0.56	0.05	0.56
40	10.01	2.73	1.91	1.61	0.2	0.74	0.82	0.74

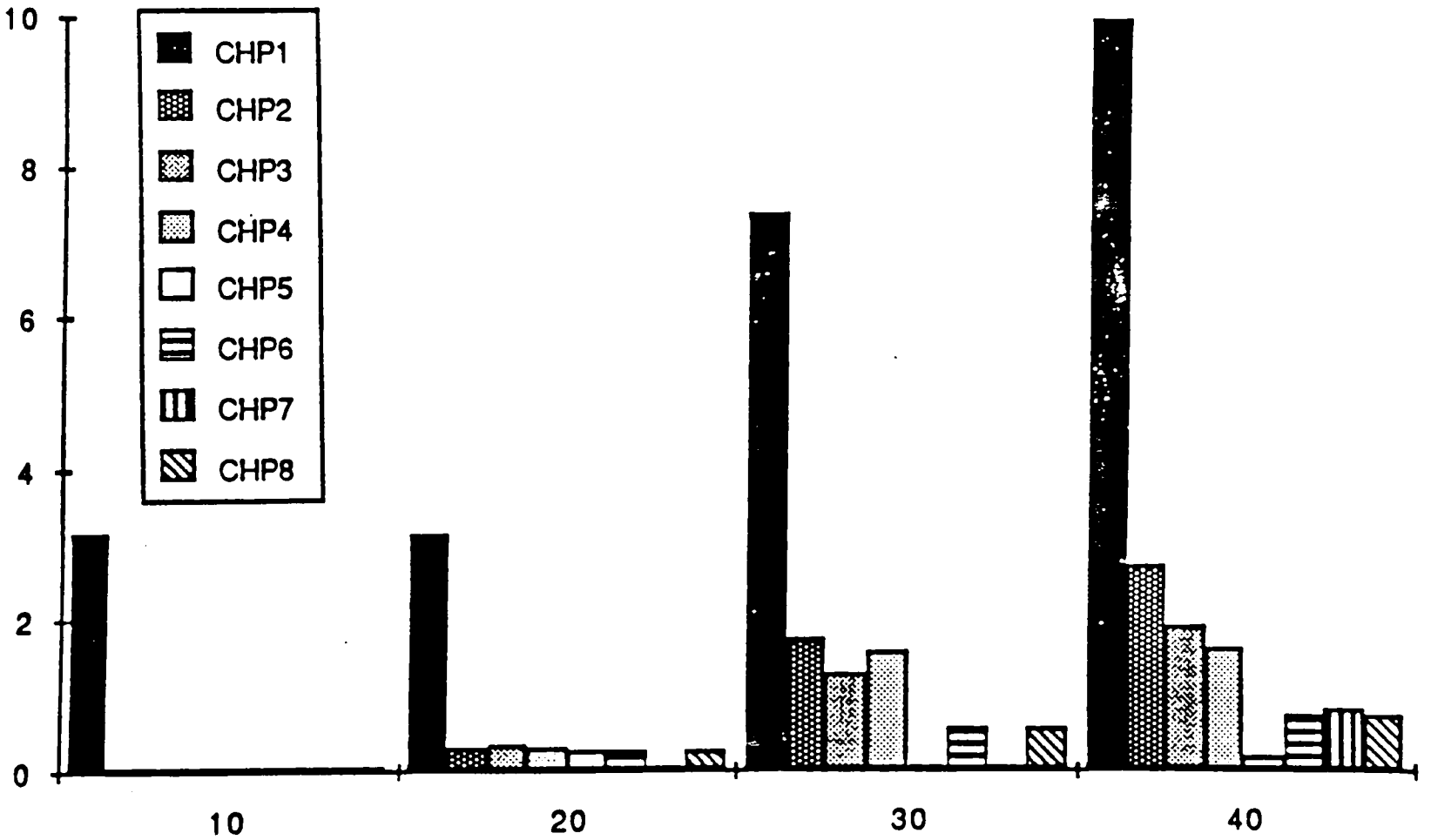


Figure 15: The Trend of Average Performance in (HPO=AB/N≤NF/SPT=BE)

CHPA $\leq$ CHPB: the Bth composite heuristic procedure is at least as good as the Ath composite heuristic procedure.

It can be concluded that if there is a composite heuristic procedure which outperforms the other composite heuristic procedures, then this procedure must be one of the 5th, 6th, 7th or 8th composite heuristic procedures.

From Table 7, it is clear that the four procedures performed equally well in boards of 10 components, and all reached the optimal solution.

To statistically compare the performance of the four procedures in boards of 20, 30, and 40 components, three nonparamatric Friedman tests were applied (8).

The first Friedman test was applied to test the null hypothesis:

Ho: The four procedures have equal performance in boards of 20 components.

The alternative hypothesis is:

H1: At least one of the procedures tends to yield better performance than at least one other procedures in boards of 20 components.

The second Friedman test was applied to test the null hypothesis:

Ho: The four composite heuristic procedures have equal performance in boards of 30 components.

The alternative hypothesis is:

H1: At least one of the procedures tends to yield better performance than at least one other procedures in boards of 30 components.

The third Fieldman test was applied to test the null hypothesis:

Ho: The four composite heuristic procedures have equal performance in boards of 40 components.

The alternative hypothesis is:

H1: At least one of the procedures tends to yield better performance than at least one other procedures in boards of 40 components.

Let Rij be rank (from 1 to 4) assigned to heuristic j (j=CHP5, CHP6, CHP7, CHP8) on boards i. The best composite heuristic procedure is assigned a rank of 1. In case of ties, average ranks are used. Next, Rj's are calculated. The first part of Table 9 summarizes the calculations required by the Friedman test for boards of 20 components. The second part of Table 9 summarizes the calculations required by the Friedman test for boards of 30 components. The third part of Table 9 summarizes the calculations required by the Friedman test for boards of 40 components.

For boards of 20 components, the test statistic T2 has a value of 1. For boards of 30 components, the test

TABLE 9

Calculations Required for Friedman Tests:
(HPO=AB/N≤NF/SPT=BE)

N	Board no.	RANK			
		CHP5	CHP6	CHP7	CHP8
20	6	2.5	2.5	2.5	2.5
20	7	2.5	2.5	2.5	2.5
20	8	2.5	2.5	2.5	2.5
20	9	2.5	2.5	2.5	2.5
20	10	3	3	1	3
Rj		13	13	11	13
(Rij)*(Rij)		34	34	26	34
i					
A2		128			
B2		125.6			
Test Statistic T2		1			
Critical Region		T2 > 3.49 at alpha = 0.05			
30	11	1.5	3.5	1.5	3.5
30	12	2.5	2.5	2.5	2.5
30	13	2.5	2.5	2.5	2.5
30	14	2.5	2.5	2.5	2.5
30	15	2.5	2.5	2.5	2.5
Rj		11.5	13.5	11.5	13.5
(Rij)*(Rij)		27.25	37.25	27.25	37.25
i					
A2		129			
B2		125.8			
Test Statistic T2		1			
Critical Region		T2 > 3.49 at alpha = 0.05			
40	16	1	2.5	4	2.5
40	17	1.5	3.5	1.5	3.5
40	18	3.5	1.5	3.5	1.5
40	19	3.5	1.5	3.5	1.5
40	20	1	2.5	4	2.5
Rj		10.5	11.5	16.5	11.5
(Rij)*(Rij)		28.75	29.25	58.75	29.25
i					
A2		146			
B2		129.4			
Test Statistic T2		1.1			
Critical Region		T2 > 3.49 at alpha = 0.05			

statistic T2 has a value 1, and for boards of 40 components, the test statistics has a value of 1.1. The critical region at alpha =0.05 corresponds to all values of T2 greater than 3.49. Therefore, in all 3 tests, we accept the null hypothesis. That is, the four composite heuristic procedures examined have equal performance.

Computational Results: (HPO=AB/N≤NF/SPT=AC)

It is known that elimination of infeasible assembly steps in this assembly environment class can eliminate possibility of board delay while Jumping Assignment Method can be applied to obtain zero pick delay. Thus, zero total delay is always reached. In Table 10, the performance of the eight different composite heuristic procedures on total X-Y table travel time in assembly environment class (HPO=AB/N≤NF/SPT=AC) are compared against the best known solutions. Note that there are great similarities between the results of Table 7 and Table 10. The reason is that most of the heuristic procedures reached the same assembly sequence on the same board under the two assembly environment classes. This implies that reducing accuracy capability by about 39% will not affect the near-optimal assembly plan.

TABLE 10

Computational Results: (HPO=AB/N≤NF/SPT=AC)

Board No.	N	BEST Known Solution	Composite Heuristic Procedures 1 (per-cent over)	2 (per-cent over)	3 (per-cent over)	4 (per-cent over)	5 (per-cent over)	6 (per-cent over)	7 (per-cent over)	8 (per-cent over)	Overall CPU (sec.)
1	10	1582	0	0.95	0	0	0	0	0	0	0.20
2	10	2234	5.86	0	0	0	0	0	0	0	0.21
3	10	2186	3.06	2.56	0	0	0	0	0	0	0.21
4	10	1996	6.86	0	0	0	0	0	0	0	0.20
5	10	1983	0	0	0	0	0	0	0	0	0.19
6	20	1966	0	0	0	0	0	0	0	0	0.96
7	20	2384	3.64	0	0	0	0	0	0	0	0.86
8	20	2718	1.10	0	0	0	0	0	0	0	0.84
9	20	2558	3.52	0.	0	0	0	0	0	0	0.89
10	20	2618	7.45	1.45	1.72	1.45	1.45	1.45	0	1.45	0.92
11	30	3516	2.90	2.33	0.26	2.33	0.26	0	0.26	0	3.16
12	30	3354	13.42	1.41	0	0	0	0	0	0	3.38
13	30	3141	1.88	2.20	0	1.72	0	0	0	0	3.42
14	30	3137	16.16	0.96	5.90	0.96	0	0	0	0	3.37
15	30	3093	5.24	1.42	0.32	1.42	0	0	0	0	3.42
16	40	3765	17.42	5.21	3.64	1.62	0	0.61	2.36	0.61	8.31
17	40	3538	10.34	5.69	1.07	2.34	0	2.22	0	2.22	8.92
18	40	3541	8.13	3.95	1.80	3.64	0.45	0.45	0.45	0.45	8.88
19	40	3593	7.27	0	0.56	0	0.56	0	0.56	0	8.69
20	40	4016	6.90	2.04	2.48	2.04	0	2.04	0.75	2.04	7.66

The average performance on boards of 10, 20, 30 and 40 components are summarized in Table 11. Figure 16 depicts the trend of these average performance. It can be seen from Figure 16 that the average performance of each of the eight composite heuristic procedures declines as the number of components increases. As in (HPO=AB/N≤NF/SPT=BE), nearest neighbor method is the worst.

To statistically compare the performance of the last four composite heuristic procedures in boards of 20 components, 30 components and 40 components, three Friedman tests were applied. The calculations required are summarized in Table 12. In all three tests, the null hypothesis is not rejected. That is, the four composite heuristic procedures examined have equal performance.

Computational Results: (HPO=A/N≤NF/SPT=RT)

In previous chapters it was found that some assembly steps may generate board delay even if infeasible assembly steps are excluded from the assembly sequence. Fortunately, the Jumping Assignment method can still be applied to guarantee zero pick delay. From data collected in this experiment, it is found that zero board delay was reached by all composite heuristic procedures except the nearest

TABLE 11

Average Performance: (HPO=AB/N≤NF/SPT=AC)

Number of Components on Board	Composite Heuristic Procedures							
	CHP1 (per-cent over)	CHP2 (per-cent over)	CHP3 (per-cent over)	CHP4 (per-cent over)	CHP5 (per-cent over)	CHP6 (per-cent over)	CHP7 (per-cent over)	CHP8 (per-cent over)
10	3.07	0.70	0	0	0	0	0	0
20	3.14	0.29	0.34	0.29	0.29	0.29	0	0.29
30	7.92	1.67	1.30	1.29	0.05	0	0.05	0
40	10.01	3.38	1.91	1.93	0.2	1.06	0.82	1.06

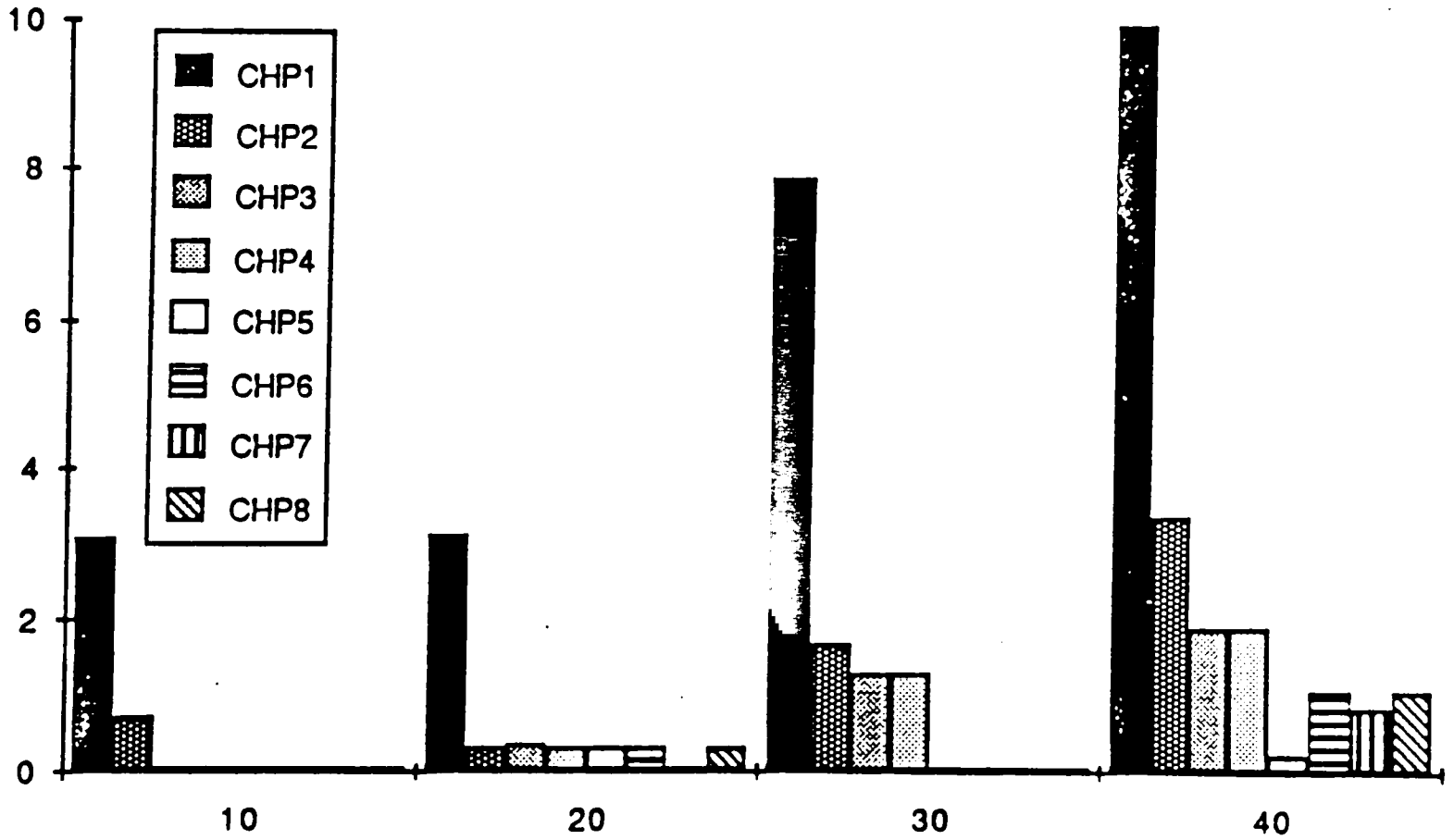


Figure 16: The Trend of Average Performance in (HPO=AB/N≤NF/SPT=AC)

TABLE 12

Calculations Required for Friedman Tests:
(HPO=AB/N≤NF/SPT=AC)

N	Board no.	RANK			
		CHP5	CHP6	CHP7	CHP8
20	6	2.5	2.5	2.5	2.5
20	7	2.5	2.5	2.5	2.5
20	8	2.5	2.5	2.5	2.5
20	9	2.5	2.5	2.5	2.5
20	10	3	3	1	3
Rj		13	13	11	13
(Rij)*(Rij)		34	34	26	34
i					
A2		128			
B2		125.6			
Test Statistic T2		1			
Critical Region		T2 > 3.49 at alpha = 0.05			
30	11	3.5	1.5	3.5	1.5
30	12	2.5	2.5	2.5	2.5
30	13	2.5	2.5	2.5	2.5
30	14	2.5	2.5	2.5	2.5
30	15	2.5	2.5	2.5	2.5
Rj		13.5	11.5	13.5	11.5
(Rij)*(Rij)		37.25	27.25	37.25	27.25
i					
A2		129			
B2		125.8			
Test Statistic T2		1			
Critical Region		T2 > 3.49 at alpha = 0.05			
40	16	1	2.5	4	2.5
40	17	1.5	3.5	1.5	3.5
40	18	2.5	2.5	2.5	2.5
40	19	3.5	1.5	3.5	1.5
40	20	1	3.5	2	3.5
Rj		9.5	13.5	13.5	13.5
(Rij)*(Rij)		22.75	39.25	40.75	39.25
i					
A2		142			
B2		127.4			
Test Statistic T2		0.658			
Critical Region		T2 > 3.49 at alpha = 0.05			

neighbor method. The board delay in nearest neighbor method is negligible (less than 0.5%). In Table 13, the performance of the eight different composite heuristic procedures on total X-Y table travel time are compared against best known solutions. The average performance on boards of 10 components, 20 components, 30 components and 40 components are summarized in Table 14. Figure 17 depicts the trend of average performance. As usual, the average performance of each of the eight composite heuristic procedures declines as the number of components increases; the nearest neighbor method is the worst. To statistically compare the performance of the last four composite heuristic procedures in boards of 20 components, 30 components and 40 components, three Friedman tests were applied. The calculations required are summarized in Table 15.

For boards of 30 components, the test statistic T2 has a value of 3.857 which is greater than the critical value 3.49 at alpha=0.05 level. The null hypothesis, all four composite heuristic procedures are equally accurate at alpha=0.05, is rejected. It is necessary to to perform further analysis to compare individual composite heuristic procedures.

Heuristic i and heuristic j are considered different if the following inequally is satisfied:

TABLE 13

Computational Results: (HPO=A/N≤NF/SPT=RT)

Board No.	N	BEST Known Solution	Composite Heuristic Procedures 1 (per-cent over)	2 (per-cent over)	3 (per-cent over)	4 (per-cent over)	5 (per-cent over)	6 (per-cent over)	7 (per-cent over)	8 (per-cent over)	Overall CPU (sec.)
1	10	1582	0	0	0	0	0	0	0	0	0.21
2	10	2234	5.86	3.67	0	0	0	0	0	0	0.25
3	10	2186	3.57	2.56	0	0	0	0	0	0	0.22
4	10	1996	6.86	0	0	0	0	0	0	0	0.23
5	10	1983	0	9.03	0	0	0	0	0	0	0.27
6	20	1966	0	0	0	0	0	0	0	0	1.04
7	20	2384	3.64	0	0	0	0	0	0	0	0.98
8	20	2718	1.10	1.10	0	0	0	0	0	0	0.97
9	20	2558	3.52	1.99	0	0.08	0	0	0	0	1.09
10	20	2618	20.97	1.45	6.42	1.45	0	1.45	1.45	1.45	1.07
11	30	3516	2.90	3.73	0.26	3.44	0.26	2.82	0.26	2.82	3.71
12	30	3354	15.42	3.82	0	0	0	0	0	0	3.73
13	30	3141	1.88	3.79	0	1.72	0	1.53	0	0	3.71
14	30	3137	16.16	0.96	5.90	0.96	0	0	0	0	3.70
15	30	3093	5.24	5.46	0.32	4.78	0	2.78	0	2.78	3.71
16	40	3765	17.42	6.00	3.64	1.72	0	0	2.36	0	9.42
17	40	3538	10.34	4.28	1.07	2.34	0	2.22	0	2.22	10.28
18	40	3541	8.13	6.72	5.95	4.40	3.64	0	3.64	3.64	9.32
19	40	3593	7.27	2.39	0.56	0.89	0.56	0.75	0.56	0.75	8.88
20	40	4016	6.90	6.38	2.48	2.12	0	2.12	0.75	2.12	8.38

TABLE 14

Average Performance: (HPO=A/N≤NF/SPT=RT)

Number of Components on Board	Composite Heuristic Procedures CHP1 (per-cent over)	CHP2 (per-cent over)	CHP3 (per-cent over)	CHP4 (per-cent over)	CHP5 (per-cent over)	CHP6 (per-cent over)	CHP7 (per-cent over)	CHP8 (per-cent over)
10	3.26	3.05	0	0	0	0	0	0
20	5.85	0.91	1.28	0.31	0	0.29	0.29	0.29
30	7.92	3.55	1.30	2.18	0.05	1.43	0.05	1.12
40	10.01	5.15	2.34	2.29	0.84	1.02	1.46	1.75

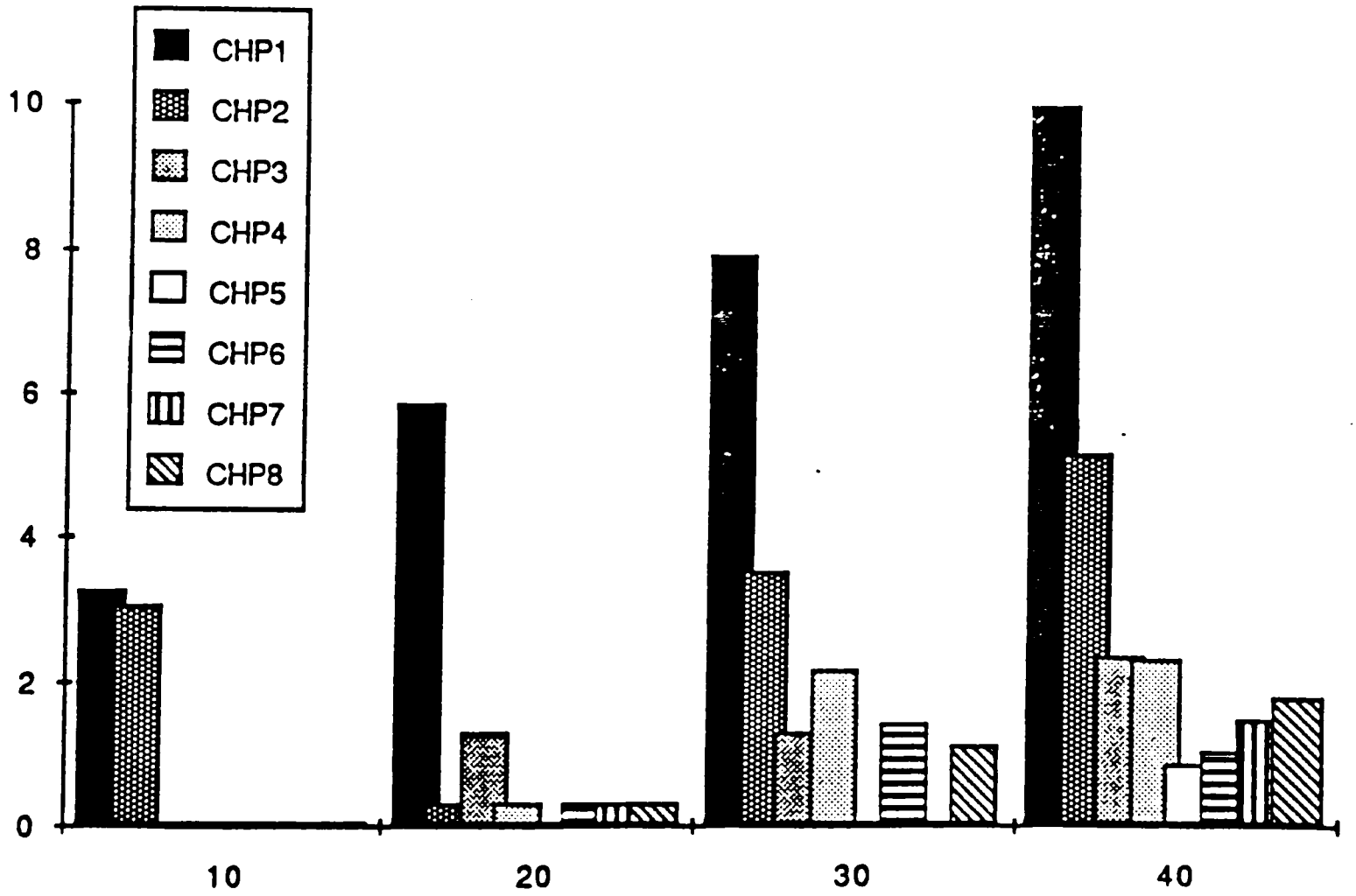


Figure 17: The Trend of Average Performance in (HPO=A/N≤NF/SPT=RT)

TABLE 15

Calculations Required for Friedman Tests:
(HPO=A/N≤NF/SPT=RT)

N	Board no.	RANK			
		CHP5	CHP6	CHP7	CHP8
20	6	2.5	2.5	2.5	2.5
20	7	2.5	2.5	2.5	2.5
20	8	2.5	2.5	2.5	2.5
20	9	2.5	2.5	2.5	2.5
20	10	1	3	3	3
Rj		11	13	13	13
(Rij)*(Rij)		26	34	34	34
i					
A2		128			
B2		125.6			
Test Statistic T2		1			
Critical Region		T2 > 3.49 at alpha = 0.05			
30	11	1.5	3.5	1.5	3.5
30	12	2.5	2.5	2.5	2.5
30	13	2	4	2	2
30	14	2.5	2.5	2.5	2.5
30	15	1.5	3.5	1.5	3.5
Rj		10	16	10	14
(Rij)*(Rij)		21	53	21	41
i					
A2		136			
B2		130.4			
Test Statistic T2		3.857			
Critical Region		T2 > 3.49 at alpha = 0.05			
40	16	1	2.5	4	2.5
40	17	1.5	3.5	1.5	3.5
40	18	2.5	2.5	2.5	2.5
40	19	3.5	1.5	3.5	1.5
40	20	1	3.5	2	3.5
Rj		9.5	13.5	13.5	13.5
(Rij)*(Rij)		22.75	39.25	40.75	39.25
i					
A2		142			
B2		127.4			
Test Statistic T2		0.658			
Critical Region		T2 > 3.49 at alpha = 0.05			

$$|Ri - Rj| > t(1-alpha/2) * \left(\frac{2*b*(A2-B2)}{(b-1)*(k-1)}\right)^{0.5}$$

where Ri, Rj, A2 and B2 are given in Table 13 and where t(1-alpha/2) is the 1-alpha/2 quantile of t distribution with (b-1)*(k-1) degrees of freedom.

The critical value for t.975 with 12 degrees of freedom is 2.179 and

$$t(0.975) * \left(\frac{2*b*(A2-B2)}{(b-1)*(k-1)}\right)^{0.5} = 4.707$$

Any two heuristics whose rank sums are more than 4.707 units apart may be regarded as being unequal. Therefore, the fifth heuristic procedure and the seventh heuristic procedure are considered better than the sixth heuristic procedure. No other differences are significant.

For boards of 20 components and 40 components, the null hypothesis is not rejected. That is, the four composite heuristic procedures have equal performance.

Computational Results: (HPO=B/N≤NF/SPT=RT)

From data collected in this experiment, it is found that zero board delay was reached by all composite heuristic procedures except nearest neighbor method. The board delay

in nearest neighbor method is negligible (less than 0.5%). In Table 16, the performance of the eight different composite heuristic procedures on total X-Y table travel time are compared against best known solutions. Note that the results in Table 7 are identical to those in Table 16. There is no need to perform statistical analysis. The data results imply that usually the assembly sequence with minimal X-Y table travel is unique. Thus, the assembly sequence which minimizes total X-Y table travel time and then minimizes assembly cycle time can be obtained by only minimizing total X-Y table travel time.

Evaluation Summary: (*/N≤NF/*)

Overall, according to the results of this empirical study, the composite heuristic procedures CHP5, CHP6, CHP7 and CHP8 can be considered as the strongest composite heuristic procedures. Average effectivness of the four strongest composite heuristic procedures is summarized in Table 17. Total X-Y table travel time obtained is no more than 1% above the best known solutions while delay is zero.

TABLE 16

Computational Results: (HPO=B/N≤NF/SPT=RT)

Board No.	N	BEST Known Solution	Composite Heuristic Procedures 1 (per-cent over)	2 (per-cent over)	3 (per-cent over)	4 (per-cent over)	5 (per-cent over)	6 (per-cent over)	7 (per-cent over)	8 (per-cent over)	Overall CPU (sec.)
1	10	1582	0	0	0	0	0	0	0	0	0.20
2	10	2234	5.86	0	0	0	0	0	0	0	0.21
3	10	2186	3.06	0	0	0	0	0	0	0	0.21
4	10	1996	6.86	0	0	0	0	0	0	0	0.26
5	10	1983	0	0	0	0	0	0	0	0	0.26
6	20	1966	0	0	0	0	0	0	0	0	0.98
7	20	2384	3.64	0	0	0	0	0	0	0	0.86
8	20	2718	1.10	0	0	0	0	0	0	0	0.84
9	20	2558	3.52	0.08	0	0.08	0	0	0	0	0.88
10	20	2618	7.45	1.45	1.72	1.45	1.45	1.45	0	1.45	0.89
11	30	3516	2.90	2.82	0.26	2.82	0.26	2.82	0.26	2.82	3.06
12	30	3354	13.42	1.61	0	0	0	0	0	0	3.30
13	30	3141	1.88	0.86	0	0	0	0	0	0	3.31
14	30	3137	16.16	0.96	5.90	0.96	0	0	0	0	3.28
15	30	3095	5.24	2.42	0.32	2.42	0	0	0	0	3.32
16	40	3765	17.42	3.85	3.64	3.11	0	0.61	2.36	0.61	8.64
17	40	3538	10.34	2.34	1.07	2.34	0	2.22	0	2.22	8.54
18	40	3541	8.13	4.77	1.80	1.61	0.45	0	0.45	0	8.32
19	40	3593	7.27	0.31	0.56	0.31	0.56	0.19	0.56	0.19	7.62
20	40	4016	6.90	2.39	2.48	0.70	0	0.70	0.75	0.70	8.18

TABLE 17

Performance Summary: (*/N≤NF/*)

Assembly Environment Class	N = 10 TL	N = 10 D	N = 20 TL	N = 20 D	N = 30 TL	N = 30 D	N = 40 TL	N = 40 D
(HPO=AB/N≤NF/SPT=BE)	0	0	0.22	0	0.31	0	0.63	0
(HPO=AB/N≤NF/SPT=AC)	0	0	0.22	0	0.03	0	0.79	0
(HPO=A/N≤NF/SPT=RT)	0	0	0.22	0	0.66	0	1.27	0
(HPO=B/N≤NF/SPT=RT)	0	0	0.22	0	0.31	0	0.63	0

where

TL: percentage of total X-Y table travel time above best known solution

D: percentage of assembly cycle time above best known solution

Experiment II: Performance Evaluation in (HPO=B/N>NF/*)

Three experiments were conducted in this phase in order to evaluate the performance of the integrated heuristic method in assembly environment classes (HPO=B/N>NF/*). Experiment II.1 includes examination of the method with IP Stage II on the first 20 boards which are of 6 component types. Experiment II.2 includes examination of the method with Quick Stage II on the same boards. Experiment II.3 extends examination to boards 21 to 35 which are of 9 component types. Comparison of Quick Stage II and IP Stage II was also made.

IP STAGE II was solved by the LINDO computer package on a VAX 11/780 computer. Quick Stage II was coded in FORTRAN and run on a VAX 11/780 computer in order to make meaningful comparison on the efficiency of the two procedures.

In these assembly environment classes, the integrated heuristic method seeks the assembly sequence with minimal X-Y table travel time at Stage I and then obtains the assignment with minimal delay with respect to that assembly sequence. Consequently, the assembly sequence obtained in (HPO=B/N>NF/SPT=BE) will be the same as that in (HPO=AB/N≤NF/SPT=BE). The assembly sequence obtained in (HPO=B/N>NF/SPT=AC) will be the same as that in (HPO=AB/N≤NF/SPT=AC).

The differences between (HPO=B/N>NF/SPT=RT) and (HPO=B/N>NF/SPT=BE) in this experiment are in the robot normal round trip time (RT) and unit delivery time of the feeder carrier (UT). Neither will affect total X-Y table travel of an assembly sequence. Thus, the assembly sequence obtained in (HPO=B/N>NF/SPT=RT) is the same as that in (HPO=B/N>NF/SPT=BE).

According to the above explanation, the performance of the integrated heuristic method on total X-Y table travel time in assembly environment classes (HPO=B/N>NF/SPT=BE) and (HPO=B/N>NF/SPT=RT) can be seen in Table 7. The performance of the integrated heuristic method on total X-Y table travel time in assembly environment class (HPO=B/N>NF/SPT=AC) can be seen in Table 10.

Computational Results of Experiment II.1

This subsection presents three sets of computational results. The first set, second set, and third set evaluate the performance of the four strongest composite heuristic procedures with IP Stage II in assembly environment classes (HPO=B/N>NF/SPT=BE), (HPO=B/N>NF/SPT=AC) and (HPO=B/N>NF/SPT=RT), respectively.

The best known solution of total delay is obtained by solving the integer programming model of IP Stage II with respect to the assembly sequence of best known total X-Y table travel time. It is worth noting that given an assembly sequence, the result of IP Stage II is actually the optimal assignment for that particular assembly sequence.

In Table 18, the total delays of the four strongest composite heuristic procedures with IP Stage II in (HPO=B/N>NF/SPT=BE) are compared against cycle time lower bound (MAC). Generally, the four strongest composite heuristic procedures reached the same level of total delay. When they did not reach the same level of total delay, the differences were very small. This integrated heuristic method with IP Stage II and any of the four strongest heuristic procedures in (HPO=B/N>NF/SPT=BE) had average total delays of 0.36%, 4.23%, 3.03%, 3.87% of the cycle time lower bound in boards of 10, 20, 30 and 40 components, respectively. On the other hand, best known solution also generated delays of 0.36%, 3.82%, 2.77%, 3.91% of cycle time lower bound (MAC).

In Table 19, the total delays of the four strongest composite heuristic procedures with IP Stage II in (HPO=B/N>NF/SPT=AC) are compared against cycle time lower bound. Note that there is great similarity between Table 18

TABLE 18

Computational Results: (HPO=B/N>NF/SPT=BE) Under IP Stage II

Board No.	N	Minimal Assembly Cycle without Delay (MAC)	Strongest Heuristics CHP5 (% of MAC)	CHP6 (% of MAC)	CHP7 (% of MAC)	CHP8 (% of MAC)	Best Known Solution (% of MAC)
1	10	11000	1.82	1.82	1.82	1.82	1.82
2	10	11000	0	0	0	0	0
3	10	11000	0	0	0	0	0
4	10	11000	0	0	0	0	0
5	10	11000	0	0	0	0	0
6	20	22000	1.82	1.82	1.82	1.82	1.82
7	20	22000	8.18	8.18	8.18	8.18	8.18
8	20	22000	2.73	2.73	2.73	2.73	2.73
9	20	22000	1.82	1.82	1.82	1.82	1.82
10	20	22000	7.27	7.27	4.55	7.27	4.55
11	30	33000	3.64	3.64	3.64	3.64	2.42
12	30	33000	4.24	4.24	4.24	4.24	4.24
13	30	33000	3.03	3.03	3.03	3.03	3.03
14	30	33000	3.03	3.03	3.03	3.03	3.03
15	30	33000	1.21	1.21	1.21	1.21	1.21
16	40	44000	2.73	2.73	2.73	2.73	2.73
17	40	44000	4.55	4.09	4.55	4.09	4.55
18	40	44000	5.45	4.55	5.45	4.55	4.55
19	40	44000	5	5.91	5	5.91	5.91
20	40	44000	1.82	1.82	1.82	1.82	1.82

TABLE 19

Computational Results: (HPO=B/N>NF/SPT=AC) Under IP Stage II

Board No.	N	Minimal Assembly Cycle without Delay (MAC)	Strongest CHP5 (% of MAC)	Heuristics CHP6 (% of MAC)	CHP7 (% of MAC)	CHP8 (% of MAC)	Best Known Solution (% of MAC)
1	10	11000	1.82	1.82	1.82	1.82	1.82
2	10	11000	0	0	0	0	0
3	10	11000	0	0	0	0	0
4	10	11000	0	0	0	0	0
5	10	11000	0	0	0	0	0
6	20	22000	1.82	1.82	1.82	1.82	1.82
7	20	22000	8.18	8.18	8.18	8.18	8.18
8	20	22000	2.73	2.73	2.73	2.73	2.73
9	20	22000	1.82	1.82	1.82	1.82	1.82
10	20	22000	7.27	7.27	4.55	7.27	4.55
11	30	33000	3.64	3.64	3.64	2.42	2.42
12	30	33000	4.24	4.24	4.24	4.24	4.24
13	30	33000	3.03	3.03	3.03	3.03	3.03
14	30	33000	3.03	3.03	3.03	3.03	3.03
15	30	33000	1.21	1.21	1.21	1.21	1.21
16	40	44000	2.73	2.73	2.73	2.73	2.73
17	40	44000	4.55	4.09	4.55	4.09	4.55
18	40	44000	5.45	5.45	5.45	5.45	4.55
19	40	44000	5	5.91	5	5.91	5.91
20	40	44000	1.82	1.82	1.82	1.82	1.82

and Table 19. The reason is that a composite heuristic reaches the same assembly sequence for the same board even if accuracy capability (AC) is reduced by 39% in this experiment. This integrated heuristic method in (HPO=B/N>NF/SPT=AC) can have total delays of 0.36%, 4.23%, 2.97%, 3.96% of cycle time lower bound in boards of 10, 20, 30 and 40 components, respectively.

In Table 20, the total delays of the four strongest composite heuristic procedures with IP Stage II in (HPO=B/N>NF/SPT=RT) are compared against cycle time without delay. Note that Table 18 and Table 29 are exactly the same except for the column cycle time without delay. As explained at the beginning of this section, the assembly sequences obtained in these two assembly environment classes are actually the same sequence. It happens that the delivery capability are the same in both assembly environment classes. Therefore, the integer programming models of IP Stage II for the two assembly environment classes in this experiment are actually the same model. Thus, they have identical total delay.

TABLE 20

Computational Results: (HPO=B/N>NF/SPT=RT) Under IP Stage II

Board No.	N	Minimal Assembly Cycle without Delay (MAC)	Strongest CHP5 (% of MAC)	Heuristics CHP6 (% of MAC)	CHP7 (% of MAC)	CHP8 (% of MAC)	Best Known Solution (% of MAC)
1	10	5500	1.82	1.82	1.82	1.82	1.82
2	10	5500	0	0	0	0	0
3	10	5500	0	0	0	0	0
4	10	5500	0	0	0	0	0
5	10	5500	0	0	0	0	0
6	20	11000	1.82	1.82	1.82	1.82	1.82
7	20	11000	8.18	8.18	8.18	8.18	8.18
8	20	11000	2.73	2.73	2.73	2.73	2.73
9	20	11000	1.82	1.82	1.82	1.82	1.82
10	20	11000	7.27	7.27	4.55	7.27	4.55
11	30	16500	3.64	3.64	3.64	3.64	2.42
12	30	16500	4.24	4.24	4.24	4.24	4.24
13	30	16500	3.03	3.03	3.03	3.03	3.03
14	30	16500	3.03	3.03	3.03	3.03	3.03
15	30	16500	1.21	1.21	1.21	1.21	1.21
16	40	22000	2.73	2.73	2.73	2.73	2.73
17	40	22000	4.55	4.09	4.55	4.09	4.55
18	40	22000	5.45	4.55	5.45	4.55	4.55
19	40	22000	5	5.91	5	5.91	5.91
20	40	22000	1.82	1.82	1.82	1.82	1.82

Computational Results of Experiment II.2

The objective of this experiment is to examine the performance of Quick Stage II on boards of 6 component types. Three sets of computational results are presented in this subsection. The first, second and third sets evaluate total delays of the four strongest composite heuristic procedures with Quick Stage II in asembly environment classes (HPO=B/N>NF/SPT=BE), (HPO=B/N>NF/SPT=AC) and (HPO=B/N>NF/SPT=RT), respectively.

In Table 21, the total delays from Quick Stage II in (HPO=B/N>NF/SPT=BE) are compared against cycle time lower bound (MAC). For boards of 6 component types, Quick Stage II has total delays of 0.36%, 4.23%, 3.27% and 3.93% of cycle time lower bound on boards of 10, 20 30 and 40 components, respectively.

In Table 22, the total delay resulted from Quick Stage II in (HPO=B/N>NF/AC) are compared against cycle time lower bound (MAC). For boards of 6 component types, Quick Stage II can have total delays of 0.36%, 4.23%, 3.21% and 4.02% of cycle time lower bound on boards of 10 components, 20 components 30 components and 40 components, respectively.

Performance of Quick Stage II in (HPO=B/N>NF/SPT=RT) is the same as that in (HPO=B/N>NF/SPT=BE) due to the same reason given in the previous subsection.

TABLE 21

Computational Results: (HPO=B/N>NF/SPT=BE) Under Quick Stage II

Board No.	N	Minimal Assembly Cycle without Delay (MAC)	Strongest Heuristics CHP5 (% of MAC)	CHP6 (% of MAC)	CHP7 (% of MAC)	CHP8 (% of MAC)	Best Known Solution (% of MAC)
1	10	11000	1.82	1.82	1.82	1.82	1.82
2	10	11000	0	0	0	0	0
3	10	11000	0	0	0	0	0
4	10	11000	0	0	0	0	0
5	10	11000	0	0	0	0	0
6	20	22000	1.82	1.82	1.82	1.82	1.82
7	20	22000	8.18	8.18	8.18	8.18	8.18
8	20	22000	2.73	2.73	2.73	2.73	2.73
9	20	22000	1.82	1.82	1.82	1.82	1.82
10	20	22000	7.27	7.27	4.55	7.27	4.55
11	30	33000	3.64	3.64	3.64	3.64	2.42
12	30	33000	4.24	4.24	4.24	4.24	4.24
13	30	33000	3.03	3.03	3.03	3.03	3.03
14	30	33000	4.24	4.24	4.24	4.24	3.03
15	30	33000	1.21	1.21	1.21	1.21	1.21
16	40	44000	3.18	2.73	2.73	2.73	2.73
17	40	44000	4.55	4.09	4.55	4.09	4.55
18	40	44000	5.45	4.55	5.45	4.55	4.55
19	40	44000	5.45	5.91	5.45	5.91	5.91
20	40	44000	1.82	1.82	1.82	1.82	1.82

TABLE 22

Computational Results: (HPO=B/N>NF/SPT=AC) Under Quick Stage II

Board No.	N	Minimal Assembly Cycle without Delay (MAC)	Strongest Heuristics CHP5 (% of MAC)	CHP6 (% of MAC)	CHP7 (% of MAC)	CHP8 (% of MAC)	Best Known Solution (% of MAC)
1	10	11000	1.82	1.82	1.82	1.82	1.82
2	10	11000	0	0	0	0	0
3	10	11000	0	0	0	0	0
4	10	11000	0	0	0	0	0
5	10	11000	0	0	0	0	0
6	20	22000	1.82	1.82	1.82	1.82	1.82
7	20	22000	8.18	8.18	8.18	8.18	8.18
8*	20	22000	2.73	2.73	2.73	2.73	2.73
9*	20	22000	1.82	1.82	1.82	1.82	1.82
10	20	22000	5.45	5.45	5.45	5.45	4.55
11	30	33000	3.64	3.64	3.64	2.42	2.42
12	30	33000	4.24	4.24	4.24	4.24	4.24
13	30	33000	3.03	3.03	3.03	3.03	3.03
14	30	33000	3.03	3.03	3.03	3.03	3.03
15	30	33000	1.21	1.21	1.21	1.21	1.21
16	40	44000	3.18	2.73	2.73	2.73	2.73
17	40	44000	4.55	4.09	4.55	4.09	4.55
18	40	44000	5.45	5.45	5.45	5.45	4.55
19*	40	44000	5.45	5.91	5.45	5.91	5.91
20	40	44000	1.82	1.82	1.82	1.82	1.82

Computational Results of Experiment II.3

The purpose of this subsection is to compare the performance of Quick Stage II and IP Stage II in terms of effectiveness as well as efficiency. From data collected at Stage I, there are actually 30 different assembly sequences generated by the four strongest composite heuristic procedures on the first 20 boards.

Table 23 summarizes total assembly cycle time and CPU time of IP Stage II and Quick Stage II for the 30 assembly sequences in assembly environment classes (HPO=B/N>NF/*). It can be seen that Quick Stage II reaches different assembly cycle times only in 3 out of the 30 samples. These three assembly cycle times reached by Quick Stage II are 1.18%, 0.43% and 0.44% above those reached by IP Stage II.

With similar levels of effectiveness, the efficiency of the two methods become important. Figure 18 depicts the comparison of CPU time required by the two methods. It can be seen that required CPU time for Quick Stage II is much less than that of IP Stage II. Under 6 component types, average CPU time required by Quick Stage II is only 17.01%, 31.38%, 35.13% and 34.40% of that required by IP Stage II for boards of 10, 20, 30 and 40 components, respectively.

TABLE 23

Performance Comparison of Quick Stage II and and IP Stage II on Boards of 6 Component types

Seq. no.	N	MAC	IP Stage II Cycle Time	IP Stage II CPU (sec.)	Quick Stage II Cycle Time	Quick Stage II CPU (sec.)
1	10	11000	11200	81.89	11200	5.22
2	10	11000	11000	50.97	11000	5.03
3	10	11000	11000	2.22	11000	5.22
4	10	11000	11000	9.69	11000	5.16
5	10	11000	11000	6.57	11000	5.12
6	20	22000	22400	95.37	22400	5.34
7	20	22000	23800	254.13	23800	5.21
8	20	22000	22600	136.28	22600	5.35
9	20	22000	22400	123.75	22400	5.22
10	20	22000	23600	269.42	23600	5.27
11	20	22000	23000	152.07	23000	5.95
12	30	33000	34200	186.93	34200	5.34
13	30	33000	34400	179.83	34400	5.36
14	30	33000	34000	106.58	34000	5.51
15*	30	33000	34000	108.89	34400	5.54
16	30	33000	32400	100.78	32400	5.48
17	30	33000	33400	256.56	33400	5.77
18	40	44000	44800	113.56	44800	5.66
19	40	44000	46000	245.42	46000	5.76
20	40	44000	46400	174.12	46400	5.63
21*	40	44000	46200	169.16	46400	6.11
22	40	44000	44800	143.38	44800	6.05
23	40	44000	45200	113.04	45200	5.54
24	40	44000	45800	235.09	45800	5.42
25	40	44000	46000	167.48	46000	5.54
26	40	44000	46600	191.98	46600	5.54
27	40	44000	44800	133.44	44800	5.50
28*	40	44000	45200	162.32	45400	5.95
29	40	44000	44800	166.70	44800	6.11
30	40	44000	44800	149.95	44800	5.75

* The assembly cycle times reached by the two methods are different for the sequence.

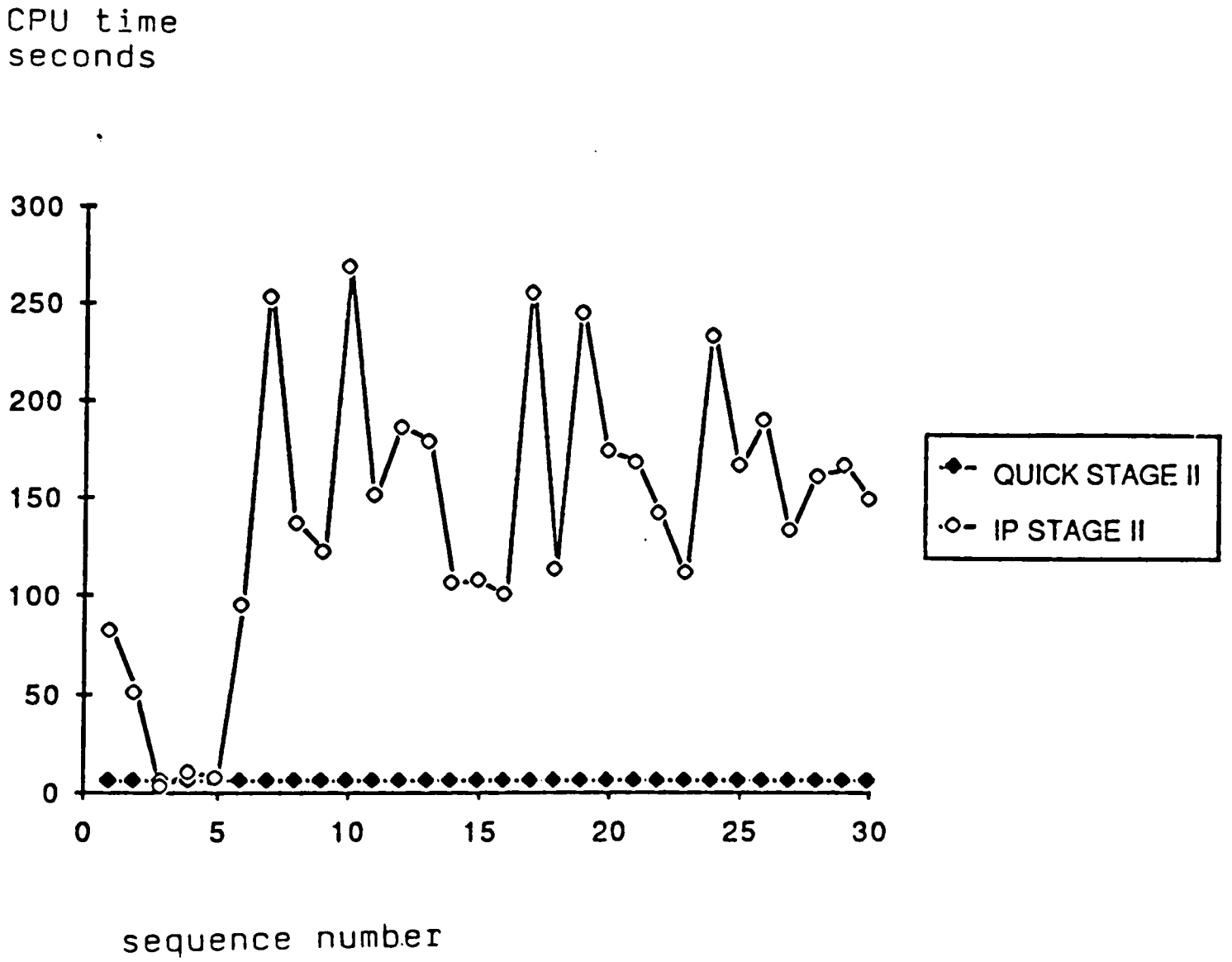


Figure 18: Efficiency Comparison of Quick Stage II and IP Stage II on Boards of 6 Component types

Comparisons of the efficiency and effectiveness of Quick Stage II against those of IP Stage II was made for boards of larger numbers of component types. An additional 15 pseudo boards of 9 component types were generated by calling GGDT subroutine of IMSL computer package. Among these 15 boards, the first 5 boards are of 20 components; the second 5 boards are of 30 components, and the third 5 boards are of 40 components. Hypothetical center 4 was selected to assemble the 15 boards. CHP8 was assumed to be the user selected composite heuristic procedure.

This experiment was performed smoothly when Quick Stage II was applied to the boards. However, tremendous CPU time was required for IP Stage II to get the optimal assignment for a single board of 9 component types. After obtaining solutions for 4 boards, it was decided not to pursue optimal assignments for the remaining boards. The effectiveness of Quick Stage II is then compared against cycle time lower bound (MAC). Table 24 summarizes the results of this experiment.

From Table 24, it can be seen that CPU time required by IP Stage II on boards of 20 components and 9 component types varys from 3331.62 seconds (55.73 minutes) to 26297.11 seconds (438.29 minutes). Meanwhile, CPU time required by Quick Stage II was around 5 to 6 seconds on boards of 20

TABLE 24

Performance Comparison of Quick Stage II and IP Stage II on Boards of 9 Component Types

Board no.	Cycle Time Lower Bound	IP Stage II Cycle Time	IP Stage II CPU (sec.)	Quick Stage II Cycle Time	Quick Stage II CPU (sec.)
21	22000	23000	3331.62	23000	5.98
22	22000	22400	4980.04	22400	5.76
23	22000	24000	26297.11	24000	5.52
24	22000	22600	7042.94	22600	5.40
25	22000	-----	--------	22800	5.54
26	33000	-----	--------	38400	6.00
27	33000	-----	--------	36000	5.90
28	33000	-----	--------	36000	5.60
29	33000	-----	--------	36600	5.26
30	33000	-----	--------	35600	5.54
31	44000	-----	--------	47400	5.97
32	44000	-----	--------	51400	5.70
33	44000	-----	--------	49200	5.84
34	44000	-----	--------	50000	6.07
35	44000	-----	--------	51800	5.83

components and 9 component types. For the 4 boards on which IP Stage II was applied, IP Stage II and Quick Stage II all reached identical assembly cycle times. This provides encouraging support for use of Quick Stage II.

For boards on which IP Stage II was not applied, the optimal assembly cycle times were not known. However, when compared against cycle time lower bound (MAC), Quick Stage II also performed well. For boards of 30 components and 9 component types, Quick Stage II reached assignments with total assembly cycle time about 10.66% above the lower bound. For boards of 40 components and 9 component types, Quick Stage II reached assignments with total assembly cycle time about 13.54% above the lower bound.

Evaluation Summary: (HPO=B/N>NF/*)

In summary, according to the computational results of the empirical study in this section, either IP Stage II or Quick Stage II can reach assembly plans with good quality. Average effectiveness of the integrated heuristic method on boards of 6 component types in assembly environment classes (HPO=B/N>NF/*) is summarized in Table 25.

For boards of 6 component types in these assembly environment classes total X-Y table travel time was consistently less than 1% above the best known solution.

TABLE 25

Performance Summary: (HPO=B/N>NF/*)

Assembly Environment Class	N=10 TL	N=10 D	N=20 TL	N=20 D	N=30 TL	N=30 D	N=40 TL	N=40 D
(HPO=B/N>NF/SPT=BE) Under Quick Stage II	0	0	0.22	0.39	0.31	0.47	0.63	0.01
(HPO=B/N>NF/SPT=BE) Under IP Stage II	0	0	0.22	0.39	0.31	0.24	0.63	-0.05
(HPO=B/N>NF/SPT=AC) Under Quick Stage II	0	0	0.22	0.39	0.03	0.41	0.79	0.11
(HPO=B/N>NF/SPT=AC) Under IP Stage II	0	0	0.22	0.39	0.03	0.18	0.79	0.04
(HPO=B/N>NF/SPT=RT) Under Quick Stage II	0	0	0.22	0.39	0.31	0.47	0.63	0.01
(HPO=B/N>NF/SPT=RT) Under IP Stage II	0	0	0.22	0.39	0.31	0.24	0.63	-0.05

where

TL: percentage of total X-Y table travel time above best known solution

D: percentage of assembly cycle time above best known solution

Total assembly cycle times reached by Quick Stage II and IP Stage II were also less than 1% above 1% above the best known assembly plans. Moreover, given the assembly sequence reached by Stage I, total assembly cycle time reached by IP Stage II was optimal for that particular sequence.

There is little difference between the total assembly cycle time reached by IP Stage II and Quick Stage II. However, the use of Quick Stage II can reduce required CPU time significantly. CPU time required by Quick Stage II is only about 30% of that required by IP Stage II for boards of 6 component types. CPU time required by Quick Stage II is only about 0.054% of that required by IP Stage II for boards of 9 component types examined. The savings increases as the number of component types increases.

Experiment III: Performance Evaluation in (HPO=A/N>NF/*)

Based on the structure designed for this integrated heuristic method, operations of this heuristic method in (HPO=B/N>NF/*) are used as Stage I and Stage II in (HPO=A/N>NF/*). From the discussion in the previous section, it is known that the same assembly sequence and assignment will be reached in both assembly environment classes (HPO=A/N>NF/SPT=BE) and (HPO=A/N>NF/SPT=RT) for the

same board. Furthermore, the pick delay matrices in Stage IV of the two classes are proportional to each other due to the same delivery capability. Hence, they will have the same satisfactory assembly plan and near-optimal assembly plan.

From the data presented in Tables 7, 10, 18 and 19, it was found that reducing accuracy capability by 39% had little impact on the near-optimal assembly plans in this experiment. Therefore, it was decided to use the performance in the assembly environment class (HPO=A/N>NF/SPT=BE) to represent the performance in the assembly environment classes (HPO=A/N>NF/*).

The best solutions were obtained by a combination of exhaustive enumeration and a heuristic procedure. Since each component type can be assigned to one and only one fedder in the feeder carrier, there are NTT!/2 possible independent assignments. By fixing each of the assignments, the 8th composite heuristic procedure was applied to obtain a local near-optimal assembly plan. Therefore, there are NTT!/2 local near-optimal solutions. The best among them is selected as the best known solution.

Two experiments are included in this phase of experimentation. Experiment III.1 evaluates the performance of the integrated heuristic method without Stage V in

assembly environment class (HPO=B/N>NF/SPT=BE). Experiment III.2 evaluates the performance of the integrated heuristic method with Stage V in assembly environment class (HPO=B/N>NF/SPT=BE).

For all board examined in this section, total delay reached is always zero. Thus, the performance measure used for this heuristic in the assembly environment class (HPO=A/N>NF/SPT=BE) is total X-Y table travel time throughout this section.

Computational Results of Experiment III.1

Table 26 summarizes the performance of the integrated heuristic method in assembly environment class (HPO=A/N>NF/SPT=BE) when Stage V is not used. The number of assignments used in Stage III are 0, 4 and 9. For boards 2 to 5, the near-optimal assembly plans are reached at the end of Stage II.

The method, when used with Stage I, IP Stage II and Stage IV, can reach assembly plans which are about 8.18% and 6.82% above best known solutions for boards of 20 and 30 components, respectively.

When four more assignments are selected by IP Stage III, the method made considerable improvement. The method

TABLE 26

Computational Results: (HPO=A/N>NF/SPT=BE) When Stage V Is Not Used

Board no	N	Best Known Solution	No. of Assignments in Stage III 0 (percent over)	4 (percent over)	9 (percent over)	CPU* (sec.)
1	10	1597	7.33(14.97)	0.81(7.33)	0.81(0)***	0.03
2	10	2234	0**	---	---	----
3	10	2186	0**	---	---	----
4	10	1996	0**	---	---	----
5	10	1983	0**	---	---	----
6	20	2204	9.17	0	0	0.23
7	20	2709	12.29	2.92(0)	2.92(0)	0.23
8	20	2831	6.04	6.04	6.04	0.22
9	20	2576	3.80(13.31)	3.80(0)	3.80(0)	0.23
10	20	3028	9.61(0.53)	0.03	0.03	0.23
11	30	3849	1.58	0	0	0.91
12	30	3759	3.88(6.54)	3.88(4.76)	0	0.93
13	30	3270	22.39	3.61	3.61	0.91
14	30	3281	2.13(19.5)	2.13(3.81)	0(3.81)	0.88
15	30	3173	4.13	0(2.77)	0(2.77)	0.89

* CPU time required by Stage IV for one assignment

** zero pick delay is reached at Stage II

*** value in parenthesis represent solution reached by Quick Stage II if it is different from that reached by IP Stage II

can reach assembly plans which are 2.67% and 1.92% above best known solution for boards of 20 and 30 components, respectively. However, when another five assignments are selected by IP Stage III the corresponding improvement is not that sharp. The method can reach assembly plans which are 1.79% and 0.72% above best known solutions.

On the other hand, the use of Quick Stage II did not affect the quality of the solutions. For boards of 20 and 30 components, the method can reach assembly plans which are 8.27% and 10.83% above the best known solutions, respectively. When four more assignments are selected by Quick Stage III, the method made encouragimg improvement. The method produced assembly plans which were 1.21% and 2.99% above the best known solutions. When another five assignments were selected by Quick Stage III, the method can reach assembly plans which are 1.21% and 2.04% above the best known solutions. This implies that the first few assignments have more added value. In summary, without Stage V, the integrated heuristic method with Stage II and Stage III can reach good assembly plans if a few more assignments are selected and added according to rules specified in the heuristic method.

It is clear that the more assignments are added into Stage IV, the better the solution is reached at the end of

Stage IV. The way of selecting the additional assignments is important. Good selection rule may reach better solutions with a lower number of additional assignments. Actually, this integrated heuristic method provides two selection rules. These are:

1. Selection rule involved in IP Stage II and IP Stage III.
2. Selection rule involved in Quick Stage II and Quick Stage III.

It is then of interest to see whether the two assignment selection rules can reach better assembly plans than if the assignments are selected randomly. In order to compare these selection rules, the Friedman Test is performed. The random selection rule can be summarized as follows:

1. The 360 independent assignments are numbered from 1 to 360.
2. For each board among boards 6 to 15, ten assignment identification numbers are selected randomly from a discrete uniform distribution with range from 1 to 360.
3. The random numbers are generated by calling GGDT subroutine of IMSL computer package.
4. For each board, the 10 assignments randomly selected are input to Stage IV, and the best assembly plan

among the 10 assignments are compared against the solution obtained by the integrated heuristic method.

The hypotheses are:

Ho: The three selection rules have equal performance.

Hl: At least one of the selection rules tends to yield better performance than at least one other assignment selection rule.

Table 27 presents total X-Y table travel times reached by the three assignment selection rules when 10 assignments are included. Calculations required by the Friedman test are summarized in Table &xlll..

The test statistic T2 has a value of 127.70. The critical region of size alpha=0.05 corresponds to values of T2 greater than 3.34. Therefore, the null hypothesis that all three selection rules are equally effective is rejected. Next, multiple comparisons will bw made among the three selection rules.

Selection Rule i and Selection Rule j are considered different if the following inequally is satisfied:

$$|Ri - Rj| > t(1-alpha/2) * \left(\frac{2*b*(A2-B2)}{(b-1)*(k-1)} \right)^{0.5}$$

where Ri, Rj and B2 are given in Table 28 and where t(1-alpha/2) is the 1-alpha/2 quantile of the t distribution with (b-1)*(k-1) degrees of freedom.

TABLE 27

Performance of the Three Assignment Selection Rules

Board No.	Designed Selection Rule (IP)	Designed Selection Rule (Quick)	Random Selection Rule
1	1610	1597	1674
2	2234	2234	2355
3	2186	2186	2550
4	1996	1996	2196
5	1983	1983	2155
6	2204	2204	2536
7	2788	2709	2836
8	3002	3002	3294
9	2576	2576	3003
10	3029	3029	3087
11	3849	3849	4036
12	3759	3774	4280
13	3388	3388	3805
14	3287	3406	3520
15	3173	3261	3629

TABLE 28

Calculations Required for Friedman Tests: The Three Assignment Selection Rules

N	Board no.	RANK		
		Designed Selection Rule (IP)	Designed Selection Rule (Quick)	Random Selection Rule
10	1	2	1	3
10	2	1.5	1.5	3
10	3	1.5	1.5	3
10	4	1.5	1.5	3
10	5	1.5	1.5	3
20	6	1.5	1.5	3
20	7	2	1	3
20	8	1.5	1.5	3
20	9	1.5	1.5	3
20	10	1.5	1.5	3
30	11	1.5	1.5	3
30	12	1	2	3
30	13	1.5	1.5	3
30	14	1	2	3
30	15	1	2	3
Rj		22	23	45
(Rij)*(Rij)		33.5	36.5	135
i				
A2		205		
B2		202.53		
Test Statistic T2		127.70		
Critical region T2 > 3.34 at alpha = 0.05				

The critical value of t.975 with 28 degrees of freedom is 2.048 and

$$t(0.975) * \left(\frac{2*b*(A2-B2)}{(b-1)*(k-1)} \right)^{0.5} = 3.33$$

Any two selection rules whose rank sums are more than 3.33 apart may be regarded as being unequal. Therefore, Random Selection is worse than the two designed selection rules proposed in the integrated heuristic method. There is no statistical difference between the two selection rules proposed in the integrated heuristic method.

Computational Results of Experiment III.2

Stage V is designed to improve the quality of an assembly plan from satisfactory level to near-optimal level. Table 29 summarizes the performance of the integrated heuristic method with Stage I, IP Stage II, IP Stage III and Stage IV and Stage V in assembly environment class (HPO=A/N>NF/SPT=BE) The number of assignments used in IP Stage III are 0, 4 and 9.

Under the condition in which no assignment is added in IP Stage III, addition of Stage V can improve the quality of the assembly plans from 8.18% to 5.10% above best known

TABLE 29

Computational Results: (HPO=A/N>NF/SPT=BE) When Stage V Is Used

Board No.	Best Known Solution	No. of Assignments in Stage III: 0			4			9		
		% over	% improvement	CPU (sec.)	% over	% improvement	CPU (sec.)	% over	% improvement	CPU (sec.)
1	1597	0.75	6.58	0.13	0.75	0.06	0.13	0.75	0.06	0.13
2	2234	—	—	—	—	—	—	—	—	—
3	2186	—	—	—	—	—	—	—	—	—
4	1996	—	—	—	—	—	—	—	—	—
5	1983	—	—	—	—	—	—	—	—	—
6	2204	9.17	0	0.59	0	0	0.20	0	0	0.20
7	2709	12.29	0	0.36	2.92	0	0.22	0	0	0.22
8	2831	3.53	2.51	0.28	3.53	2.51	0.28	3.53	2.51	0.28
9	2576	0	3.80	0.17	0	3.80	0.17	0	0	0.17
10	3028	0.53	9.08	0.36	0.03	0	0.31	0.03	0	0.31
11	3849	1.30	0.28	0.67	0	0	0.25	0	0	0.25
12	3759	3.88	0	0.89	3.88	0	0.89	0	0	0.67
13	3270	3.61	18.78	0.86	3.61	0	0.36	3.61	0	0.36
14	3281	2.13	0	0.26	2.13	0	0.26	0	0	0.19
15	3173	4.13	0	0.32	0	0	0.18	0	0	0.18

solutions for boards of 20 components. For boards of 30 components, the quality of the assembly plans is improved from 6.82% to 3.01% above the best known solutions.

Under the condition in which 4 more assignments are added, addition of Stage V can improve the quality of the assembly plans from 2.67% to 1.30% above best known solutions. No improvement is made for boards of 30 components. The quality of the assembly plans remains at the level of 1.92% above best known solutions.

Under the condition in which 10 assignments are added, addition of Stage V may improve the quality of the assembly plans from 1.79% to 0.712% above the best known solutions for boards of 20 components. No improvements were reached for boards 30 components. The quality of the assembly plans remains at the level of 0.72% above best known solutions.

The next step is to investigate the effectivness of Stage V when it is used with Quick Stage II and Quick Stage III. Since the results of the method with 4 and 9 assignments in Quick Stage III have reached good quality solutions, Stage V was added to the end of Quick Stage II only. Table 30 presents the results before and after the addition of Stage V.

The computation time required by Stage V was very short. In all boards examined, less than 1 CPU second was needed.

TABLE 30

Computational Results: (HPO=A/N>NF/SPT=BE) Under Quick Stage II and Stage V

Board no.	N	Best Known Solution	Before Addition (% over)	After Addition (% over)	Improvement %
1	10	1597	14.97	0.75	14.22
2	10	2234	----	----	----
3	10	2186	----	----	----
4	10	1996	----	----	----
5	10	1983	----	----	----
6	20	2204	9.17	9.17	0
7	20	2709	12.29	12.29	0
8	20	2831	6.04	3.53	2.51
9	20	2576	13.31	0	13.31
10	20	3028	0.53	0.53	0
11	30	3849	1.58	1.30	0.28
12	30	3759	6.54	3.88	2.66
13	30	3270	22.39	3.61	18.78
14	30	3281	19.50	2.13	17.37
15	30	3173	4.13	4.13	0

Performance Summary:
(HPO=A/N>NF/SPT=BE)

This section summarizes the effectiveness of the integrated heuristic method under assembly environment class (HPO=B/N>NF/SPT=BE) for different numbers of components on boards.

Table 31 summarizes the performance of the integrated heuristic method in assembly environment class (HPO=A/N>NF/SPT=BE) with different utilization of stages. The primary objective function value--total delay always has a value of 0. The secondary performance measurer--total X-Y table travel varies between 0% and 10.83% above the best known solutions.

The quality of the resultant total X-Y table travel depends on several variables, including the number of components in the board, the number of component types in the board, the layout distribution of component types in the board and the use of stages of the integrated heuristic method. From data collected in this experiment, the quality of the resultant assembly plans can be improved to less than 3% above the best known solutions by adding four more assignments to Stage IV.

Another way of improving the quality of assembly plans in assembly environment classes (HPO=A/N>NF/*) is through

TABLE 31

Summary of Performance in (HPO=A/N>NF/SPT=BE)

Usage of Stage V	Stage II and Stage III	N	No. of Assignments in Stage III: 0 TL	0 D	4 TL	4 D	9 TL	9 D
NO	IP	10	1.47	0	0.16	0	0.16	0
NO	Quick	10	2.99	0	0.16	0	0	0
NO	IP	20	8.18	0	2.67	0	1.79	0
NO	Quick	20	8.27	0	1.21	0	1.21	0
NO	IP	30	6.82	0	1.92	0	0.72	0
NO	Quick	30	10.83	0	2.99	0	2.04	0
YES	IP	10	0.15	0	0.15	0	0.15	0
YES	Quick	10	0.15	0	----	--	----	-
YES	IP	20	5.10	0	1.30	0	0.71	0
YES	Quick	20	5.10	0	----	--	----	-
YES	IP	30	3.01	0	1.92	0	0.72	0
YES	Quick	30	3.01	0	----	--	----	-

where

TL: percentage of total X-Y table travel time above best known solution

D: percentage of assembly cycle time above best known solution

the use of Stage V. Without any additional assignment, Stage V can reach assembly plans whose total X-Y table travel is less than 5% above the best known solutions.

Actual Board Numerical Example

An illustration of application of this integrated heuristic method to an actual printed circuit board is given in this section. The task is to mount 109 Surface Mount Devices to a board. The 109 components belong to 47 component types. Figure 19 depicts the locations of the 109 components in the board. Figure 20 depicts the layout distribution of the component types in the board. Complete description of the board is listed in Appendix B. Related system parameters are summarized as follows:

1. N: Total number of components to be mounted is 109.
2. NTT: Total number of component types is 47.
3. NF: Total number of feeders in the feeder carrier is 60.
4. BE: Board extreme moving time is 531 milli-seconds.
5. RT: Robot normal round trip time is 904 milli-seconds.
6. AC: Accuracy capability of the X-Y table is 279 milli-seconds.

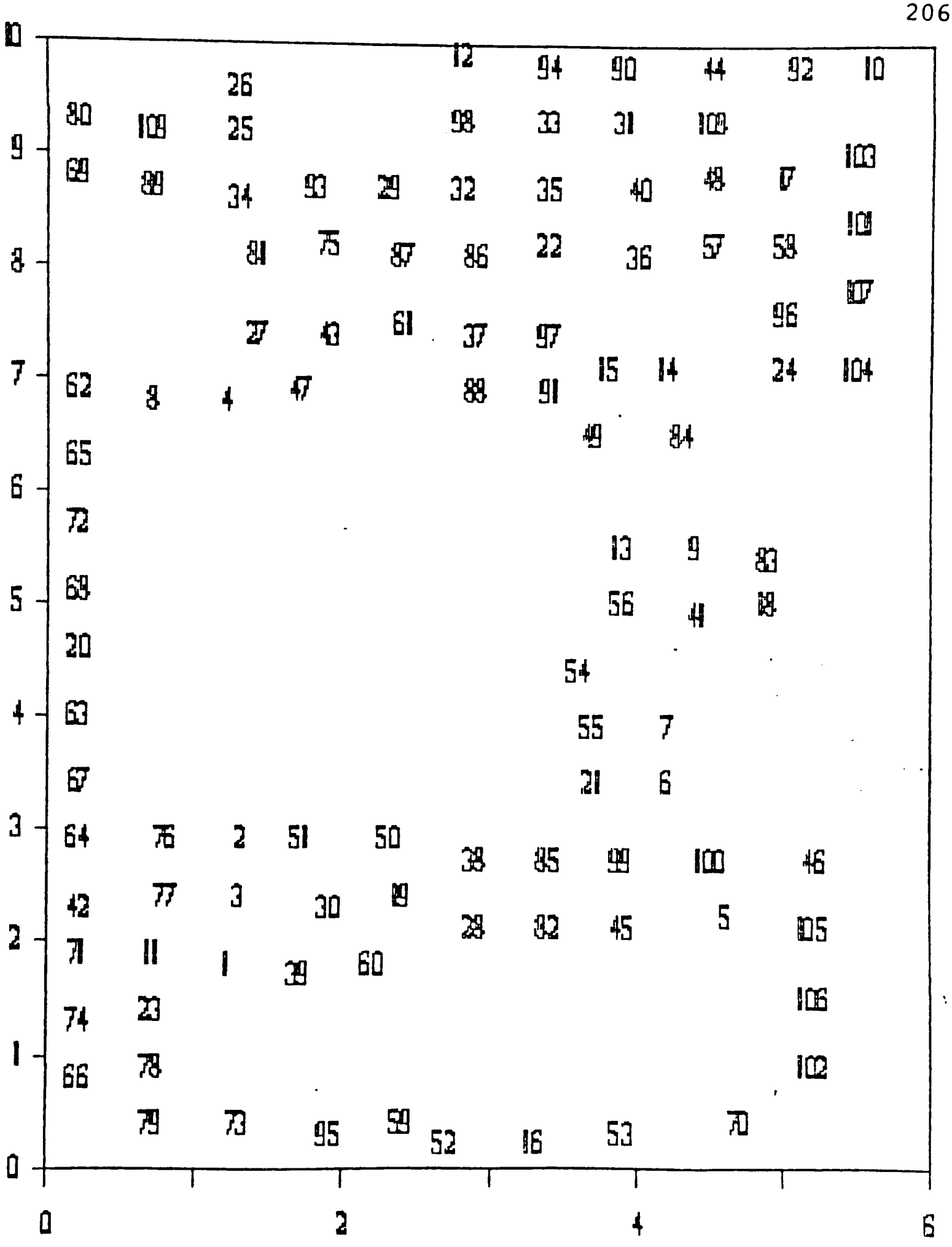


Figure 19: Layout of the 109-component Board

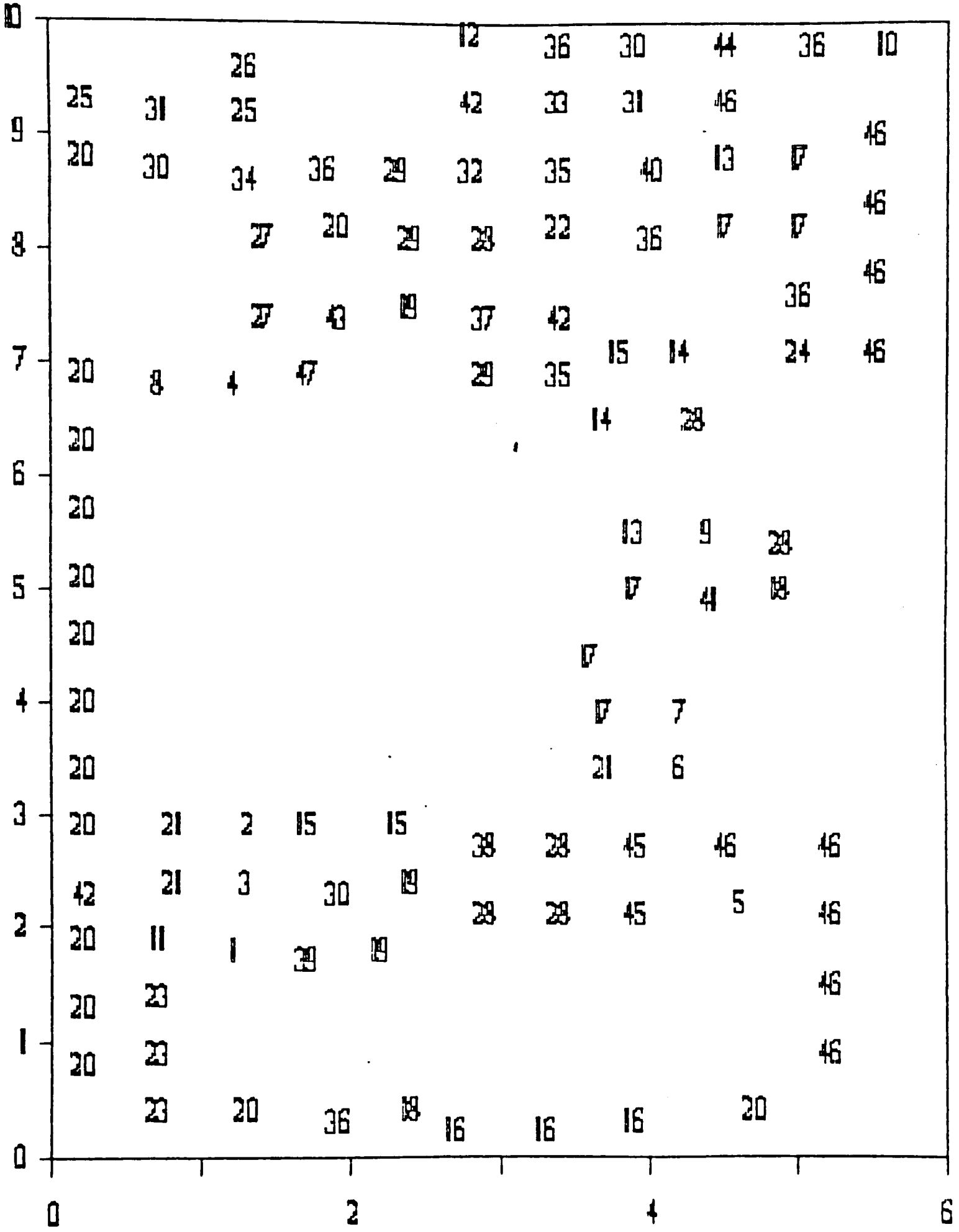


Figure 20: Layout of Component Types in the 109-component Board

7. UT: Unit delivery time of the feeder carrier is 130 milli-seconds.
8. V: Speed of the X-Y table is 546.1 mm/second.
9. PT: The time needed to pick/place a component is 108 milli-seconds.

The number of components (N) is greater than the number of component types (NTT). Moreover, the shortest placing related timing parameter (SPT) is AC with a value of 279. Depending on the priority order of the two objectives, the assembly environment is either (HPO=A/N>NF/SPT=AC) or (HPO=B/N>NF/SPT=AC). If the due date is near, the manager may want to minimize assembly cycle time first and then to minimize total X-Y table travel. The corresponding assembly environment class will then become (HPO=A/N>NF/SPT=AC). On the other hand, if shop capacity is sufficient to handle all the orders, the manager may want to save potential maintainence cost. Thus, minimizing total X-Y table travel receives higher priority than minimizing assembly cycle time. The corresponding assembly environment class is then (HPO=B/N>NF/SPT=AC).

Application of the integrated heuristic method on the board in assembly environment class (HPO=B/N>NF/SPT=AC) is presented first, followed by the application in assembly environment class (HPO=A/N>NF/SPT=AC).

Near-optimal Assembly Plan in (HPO=B/N>NF/SPT=AC)

Under assembly environment class (HPO=B/N>NF/SPT=AC), the assembly sequence reached at end of Stage I has total X-Y table travel time of 2879 milli-seconds which is equivalent to 61.90 inches of travel distance. CPU time required by Stage I is 84.73 seconds on an IBM 3081 computer. The assembly sequence is depicted in Figure 21. After the near-optimal assembly sequence is reached by Stage I, Quick Stage II is executed to obtain the "best" assignment of the component types to the feeders of the feerder carrier. The resultant assignment generated 10704 milli-seconds of delay for the near-optimal assembly sequence. The delay time is 9.70% of cycle time lower bound. Table 32 presents the near-optimal assignment obtained at the end of Stage II. CPU time required by Quick Stage II is 29.18 seconds on an IBM 3081 computer.

Near-optimal Assembly Plan in (HPO=A/N>NF/SPT=AC)

In order to find a near-optimal assembly plan for the 109-component board in assembly environment class (HPO=A/N>NF/SPT=AC), Quick Stage II and Quick Stage III are included in the integrated heuristic method. At the end of Quick Stage II, it is found that there are 23 cells, in the

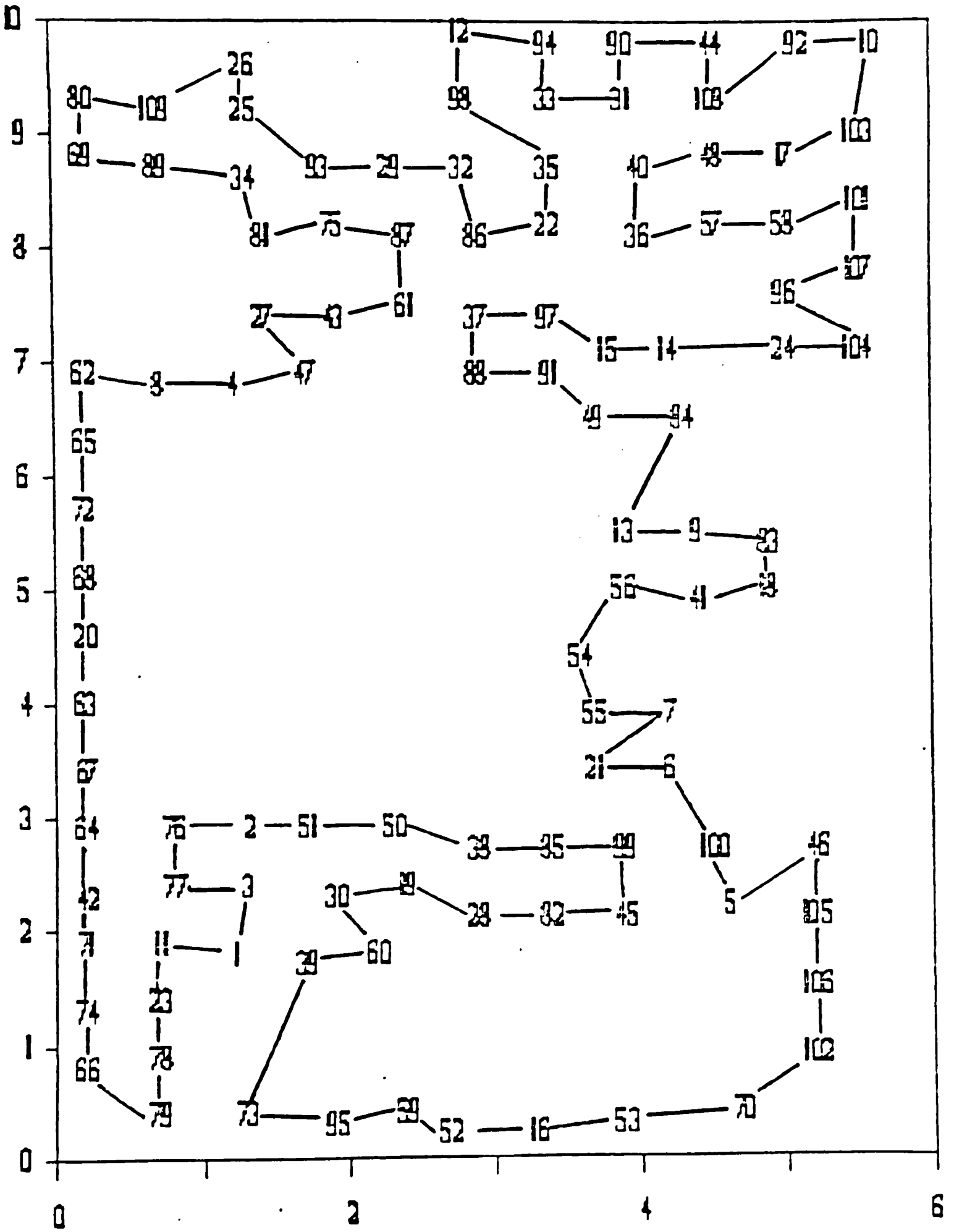


Figure 21: Near-optimal Assembly Sequence in (HPO=B/N>NF/SPT=AC)

TABLE 32

Near-optimal Assignment of Feeders in (HPO=B/N>NF/SPT=AC)

Component Type ID Number	Feeder Position	Component Type ID Number	Feeder Position
1	1	2	3
3	4	4	46
5	10	6	7
7	8	8	42
9	14	10	22
11	2	12	29
13	13	14	18
15	9	16	33
17	16	18	27
19	36	20	34
21	5	22	23
23	6	24	11
25	32	26	39
27	44	28	20
29	30	30	41
31	38	32	26
33	31	34	45
35	25	36	24
37	35	38	12
39	40	40	19
41	21	42	28
43	43	44	37
45	15	46	17
47	47		

2-switching total pick delay matrix (PDCM), having the same amount of delay time. That is, there are at least 16 additional alternative assignments which can generate the same smallest total delay, 10704 milli-seconds, for the assembly sequence obtained at Stage I. The original assignment obtained at Stage II (the assignment reported as the "best" assignment in (HPO=B/N>NF/SPT=AC)) was sent into Stage IV first, followed by 9 of the remaining alternative assignments. The resultant total X-Y table travel time of the 10 local near-optimal assembly plans and associated CPU time spent in Stage IV are summarized in Table 33.

From data shown in Table 33, it is clear that the assignment obtained at Quick Stage II resulted in the smallest X-Y table travel time of 4049 milli-seconds. No improvement is made when the other 9 assignments are entered into Stage IV. The assembly sequence under the first assignment is then treated as the near-optimal assembly sequence, and the first assignment is treated as the near-optimal assignment. The near-optimal assembly sequence is depicted in Figure 22. Table 34 shows the near-optimal asssignment.

It is of interest to investigate the influence of the priority order of the two objectives on the near-optimal assembly plan. From data presented in this section, the near-optimal assembly plan in (HPO=B/N>NF/SPT=AC) is with

TABLE 33

Results of the 10 Local Near-optimal Assembly Plans

Assignment no.	Component Types Involved in Pairwise Change	Total X-Y Travel Time (milli-sec.)	CPU at Stage IV (sec.)
1	None	4049	60.55
2	(1, 3)	4424	46.55
3	(1,11)	4301	60.99
4	(2, 3)	4049	60.11
5	(2,21)	4049	53.87
6	(3,11)	4050	51.63
7	(3,21)	4049	60.73
8	(4,34)	4049	60.17
9	(6, 7)	4049	60.65
10	(6,23)	4049	60.44

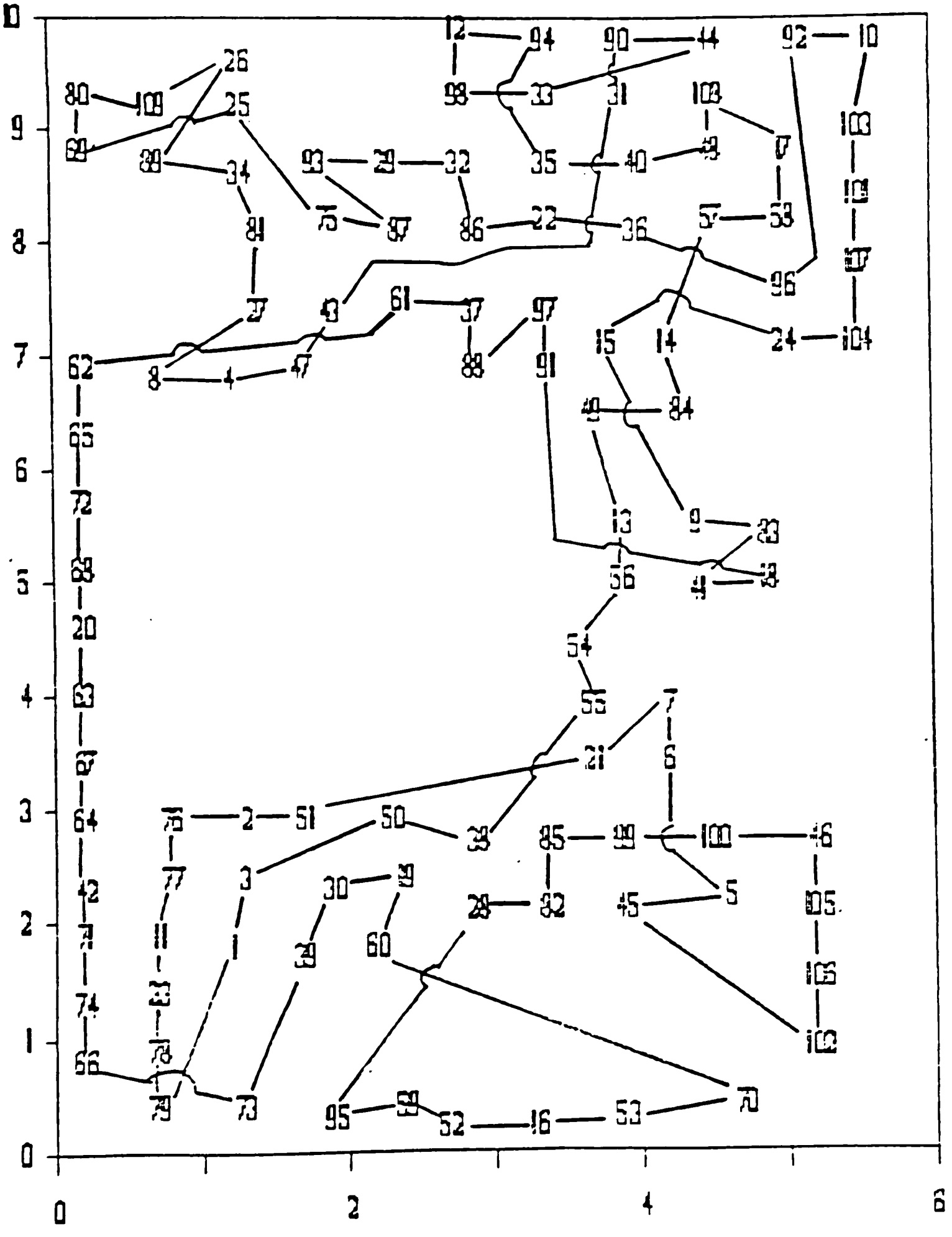


Figure 22: Near-optimal Assembly Sequence in (HPO=A/N>NF/SPT=AC)

TABLE 34

Near-optimal Assignment of Feeders in (HPO=A/N>NF/SPT=AC)

Component Type ID Number	Feeder Position	Component Type ID Number	Feeder Position
1	1	2	3
3	4	4	46
5	10	6	7
7	8	8	42
9	14	10	22
11	2	12	29
13	13	14	18
15	9	16	33
17	16	18	27
19	36	20	34
21	5	22	23
23	6	24	11
25	32	26	39
27	44	28	20
29	30	30	41
31	38	32	26
33	31	34	45
35	25	36	24
37	35	38	12
39	40	40	19
41	21	42	28
43	43	44	37
45	15	46	17
47	47		

total X-Y table trave of 2879 milli-seconds and total delay of 10704 milli-seconds. On the other other hand, the near-optimal assembly plan in (HPO=A/N>NF/SPT=AC) is travel of 4049 milli-seconds and zero delay. Treating 2879 as the minimal total X-Y table travel time needed, the manager is actually reducing the delay time (9.70% of cycle time lower bound) by sacrificing total X-Y table travel (40.64% of travel time lower bound) when the priority order of the two objectives is changed from minimizing total travel first and then minimizing assembly cycle time to minimizing assembly cycle time first and then minimizing total X-Y table travel.

CHAPTER VII

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

Summary and Conclusions

The purpose of this research was threefold. First, a set of parameters was defined for use in analyzing assembly planning problem in robotized PCB centers. Secondly, a number of mathematical models were developed to optimize assembly plans for a robotized PCB assembly center by taking considerations of the coordination of robot movements, X-Y table travel and movements of the feeder carrier. Third, an integrated heuristic method was developed to obtain near-optimal assembly plans of the robotized PCB assembly center.

The robotized PCB assembly center considered in this research consisted of a moving X-Y table, a moving feeder carrier and a pick-and-place robot. The assembly planning problem faced by the users of this robotized PCB assembly center was the one of determining the assembly sequence and the assignment of feeders in the feeder carrier to optimize two prioritized objectives: minimizing assembly cycle time and minimizing total X-Y table travel time.

A thorough analysis of the problem was conducted first. Three groups of parameters were defined. These are: (1) Non-timing parameters (2) timing parameters and (3) priority parameter. Based on the parameters defined, 10 different assembly environment classes were identified. These were:

1. (HPO=AB/N≤NF/SPT=BE)
2. (HPO=AB/N≤NF/SPT=AC)
3. (HPO=A/N≤NF/SPT=RT)
4. (HPO=B/N≤NF/SPT=RT)
5. (HPO=A/N>NF/SPT=BE)
6. (HPO=A/N>NF/SPT=AC)
7. (HPO=A/N>NF/SPT=RT)
8. (HPO=B/N>NF/SPT=BE)
9. (HPO=B/N>NF/SPT=AC)
10. (HPO=B/N>NF/SPT=RT)

Mathematical models for solving the assembly planning problems in the 10 different assembly environment classes were developed. The mathematical models in assembly environment classes in which the number of components is no less than the number of feeders in the feeder carrier were modeled as traveling salesman problems with modification of the associated cost matrices. The mathematical models for the remaining assembly environment classes were of non-linear nature and were linearized to have three modules.

These are: (1) assembly sequence module, (2) assignment module and (3) connection module. These models were more complicated than the previous ones.

An integrated heuristic method of five stages was developed to quickly reach a near-optimal assembly plan. With different combination of the stages, the assembly planning problem in different assembly environment classes can be solved. The function of the five stages can be summarized as follows:

Stage I: Find the near-optimal assembly plan for assembly environment classes in which $N \leq NF$. Find the assembly sequence with minimal X-Y table travel for assembly environment classes in which $N > NF$.

Stage II: Fix the assembly sequence obtained at Stage I and find its corresponding optimal assignment.

Stage III: Fix the assembly sequence obtained at Stage I and find an additional K best assignments.

Stage IV: Fix each of the assignments obtained at Stage II and III to find the corresponding best assembly sequence. Each assignment and assembly sequence pair constitutes a candidate of satisfactory assembly plan. Choose the best one as the satisfactory assembly plan.

Stage V: Find the near-optimal assembly plan by seeking improvement on the satisfactory assembly plan.

A set of experiments was performed to evaluate the performance of this integrated heuristic method in the 10 different assembly environment classes. For assembly environment classes in which the number of feeders is no less than the number of components, results indicates that zero delay is always guaranteed while total X-Y table travel time is less than 1% above the best known solutions. For assembly environment in which the number of components is greater than the number of feeders and total X-Y table travel receives higher priority than total assembly cycle time, the integrated heuristic method can reach a near-optimal assembly plan in which total X-Y table traveling and total assembly cycle time are no more than 1% above the best known solutions for the pseudo boards examined in this research. For assembly environment classes in which the number of components is greater than the number of feeders and total assembly cycle time receives higher priority than total X-Y table travel time, the integrated heuristic method can reach assembly plan in which total delay is always zero while total X-Y table travel time is no more than 11% above the best known solutions. The quality of solution with regard to total X-Y table travel depends on the number of components, the number of component types and the use of the different stages of the integrated heuristic method. The quality can

be improved to 3% above the best known solutions when Stage III and/or Stage V of the integrated heuristic method. An actual nemerical example was given to illustrate the application of this integrated heuristic method.

Recommendations for Future Research

This was the first study to examine assembly planning problem by taking consideration of the coordination of the robot movement, X-Y table travel and the movement of the feeder carrier. In order to solve assembly planning problems in robotized PCB assembly centers, additional studies are needed. Based on the results of this study, the following suggestions are offered for future research:

1. Efficient computer package for solving real-world large scale integer problems is needed. The first step is to develop a computer package which can secure optimal solutions in a reasonable amount of time such that evaluation of a heuristic is possible.
2. Investigation of assembly planning problem is needed for 2-robot PCB assembly center in which the robots are synchronized and two feeder carriers are involved, one on each side of the X-Y table. Due to the foreseeable complexity of this assembly environment, it is suggested that the two objectives be treated separately.

3. Research in decomposing the assembly task of a PCB board into several robotized PCB assembly centers may be helpful to the PCB manufacturers and is worth of investigation.
4. Investigation of assembly planning problem is needed for other robotized PCB assembly centers in which the robot is the only moving objects and the components of the same type are assigned to the same bin cell.
5. Investigation of potential redesign of this robotized PCB assembly center may be helpful to the equipment manufacturer. Two studies are suggested. First, investigate the possibility of increasing delivery capability of the feeder carrier. Increasing delivery capability does not necessarily require increasing the speed of the feeders carrier. One way of doing it may be to supply a number of feeder carriers on each side of the X-Y table. For example, two feeder carriers may be included in the one-robot PCB assembly center while four feeder carriers may be included in the two-robot PCB assembly center. delivery capability will be doubled in both cases. Another redesign approach worth of analysis is to provide multiple heads to the robot arm. Thus, a number of components can be mounted to the board in one cycle.

Applications of the solution procedures developed in this research are not limited to PCB assembly. Various areas may benefit from application of the solution procedures. These include: PCB drilling, automated sequencing machine setups and computer aided drawing system arrangement. Research in applying the solution procedures developed in this research to the above areas is worth of investigation.

REFERENCES

1. Baumann, W., Haensler, R., Hartmann, R., and Siemens, A., "Operating and Idle Times for Cyclic Multi-machine Serving," The Industrial Robot, March, pp. 44-49, 1981.

2. Bellman, R.E., "Dynamic Programming Treatment of the Traveling Salesman Problem," J. Assoc. Comput Math, 9, pp. 61-63, 1962.

3. Bellmore, M., Nemhauser, G., "The Traveling Salesman Problem-A Survey," Operations Research, Vol. 16, p. 538., 1968.

4. Bjorke, O., "Use Robots within Manufacturing Cells," Conference on Computer Aided Manufacturing, National Engineering Laboratory, East Kilbride, Scotland, Glasgow, 1978.

5. Chang, C. A., and Goldman, J., "An Offline Design Procedure for the Work Station of Assembly Robots," 1982 Fall Industrial Engineering Conference Proceedings.

6. Christofides, N., Eilon, S., "Algorithms for Large-Scale Traveling Salesman Problems," Oper. Res. Quart., Vol. 23, pp. 511-518, 1972.

7. Chvatal, V., "Edmonds Polytopes and Weakly Hamiltonian Graphs," Mathematical Programming, Vol. 5, pp. 29-40, 1973.

8. Conover, W. J., Practical Nonparametric Statistics, 2nd. edition, John Wiley & Sons Ltd., 1980.

9. Croes, G.A., "A Method for Solving Traveling Salesman Problems," Operations Research, Vol. 6, pp. 791-812, 1958.

10. Crowder, H., Padberg, M.W., "Solving Large-Scale Zero-One Linear Problems to Optimality," Management Sci., Vol. 26, pp. 495-509, 1980.

11. Dantzig, G.B. and Fulkerson, D.R., Johnson, S.M., "Solution of a Large-Scale Traveling Salesman

Problem," Operation Research, Vol 2, pp. 293-410, 1954.

12. Dorf, Richard C., "Robotics and the Automated Manufacturing Industry," Proceedings of the 13th International Symposium on Industrial Robots, April, 1983.

13. Drezner, Zvi, and Nof, Shimon Y., "On Optimizing Bin Picking and Insertion Plans for Assembly Robots," AIIE Transactions, Vol. 16, No. 3, 1984.

14. Eastman, W.L., "Linear Programming with Pattern Constraints," Ph.D. Thesis, Harvard University, Cambridge, MA., 1958.

15. Edmonds, J., "Maximum Matching and a Polyhedron with o, 1 vertices," Journal of Research of the National Bureau of Standard, 69B pp. 125-130, 1965.

16. Eilon, S., Watson-Gandy, C.D.T., and Christofides, N., Distribution Management, Charles Griffins, London., 1971.

17. Fisher, E. L., Nof, S. Y., and Seidmann, A., "Robot System Analysis: Basic Concepts and Surveys of Methods," Fall Industrial Engineering Conference Proceedings, 1982.

18. Golden, B .L., Bodin, L. D.,Doyle, T., and Stewart, W. Jr., "Approximate Traveling Salesman Algorithms," Operations Research, Vol. 28, pp. 694-711, 1980.

19. Golden, B. L. and Stewart W. Jr., "Empirical Analysis of Heuristics," in The Traveling Salesman Problem, edited by Lawler, E. L., Lenstra, I. K., Rinnooy Kan, A. H. G. and Shimoys, D. B., John Wiley & Sons Ltd., New York, 1985.

20. Graves, S. C., and Lamar, B. W., "An Integer Programming Procedure for Assembly System Design Problems," Operations Research, Vol. 31, No. 3, 1983.

21. Graves, S. C., and Whitney, D. E., "A Mathematical Programming Procedure for Equipment Selection and System Evaluation in Programmable Assembly," Proceedings of the 18th IEEE Conference on Decisions and Control, Ft. Lauderdale, Fla., 1979.

22. Grotschel, M., "On the Symmetric Traveling Salesman Problem: Solution of A 120-city Problem," Mathematical Programming Study, Vol. 12, pp. 61-77, 1980.

23. Grotischel, M. and Padberg, M. W., "On the Symmetric Traveling Salesman Problem, Part I and Part II," Mathematical Programming, Vol. 16, pp. 265-302, 1979.

24. Heginbotham, W. B., Dooner, M., and Case, K., "Robot Application Simulation," The Industrial Robot, Vol. 6, No. 2, 1976.

25. Held, M. and Karp, R. M., "A Dynamic Programming Approach to Sequencing Problem," SIAMJ. Appl. Math., Vol. 10, pp. 196-210. 1962.

26. Held, M. and Karp, R. M., "The Traveling Salesman Problem and Minimum Spanning Trees: Part II," Math. Programming, Vol. 6, pp. 6-25, 1971.

27. Holmquist, U., "Programmable Automatic Assembly Station with New ASEA Robot," Proceedings of the 14th International Symposium on Industrial Robots, 1984.

28. Holms, J. G., "Integrating Robots into a Machining System," 1979 Fall Industrial Engineering Conference.

29. Holms, J. G., "An Automated Robot Machining System," Proceedings of the 9th International Symposium on Industrial Robots, 1979.

30. Hong, S., "A Linear Programming Approach for the Traveling Salesman Problem," Ph.D. Thesis, Johns Hopkins University, Baltimore, MD., 1972.

31. Isaac, A.M., Turban, E., "Some Comments on the Traveling Salesman Problem," Operations Research, Vol. 17, p.543, 1969.

32. Kamali, Jila et al., "A Framework for Integrated Assembly System: Humans, Automation, and Robots," Int. J. of Prod. Res., Vol. 20, No. 4, 1982.

33. Karp, R. M., "Reducibility among Combinatorial Problems," in Complexity of Computer Computations, edited by Miller, R. E. and Thatcher, J. W., Plenum Press, New York, 1972.

34. Kondoleon, A. S., "Application of Technology--Economic Model of Assembly Techniques and Programmable Machine Configurations," M.S. Theses at MIT, Cambtidge, Massachusetts, 1976.

35. Lenstra, J., and Rinnooy Kan, A., "Some Simple Applications of the Traveling Salesman Problem," Operational Research Quarterly, 24(4), pp. 717-733, 1975.

36. Lin S., "Computer Solutions of the Traveling Salesman Problem," Bell System Tech J., Vol. 44, pp. 2245-2269, 1965.

37. Lin, S. and Kernighen, B. W., "An Effective Heuristic Algorithm for the Traveling Salesman Problem," Operations Research, Vol. 21, pp. 498-516, 1973.

38. Little, J.D.C., Murty, K.G., Sweeney, D.W., Karel, C., "An Algorithm for the Traveling Salesman Problem," Operation Research, Vol. 11, pp. 972-989, 1963.

39. Maczka, Walter J. et al., "The Challenge to Electronics Assembly," Assembly Engineering, Vol. 28, No. 5, 1985.

40. Miliotis, P., "Integer Programming Approaches to the Traveling Salesman Problem," Mathematical Programming, Vol. 10, pp. 367-378, 1976.

41. Miliotis, P., "Using Cutting Plans to Solve the Symmetric Traveling Salesman Problem," Mathematical Programming, Vol. 15, pp. 177-188, 1978.

42. Morgan, C., Robotics Planning and Implementation, IFS Ltd., England, 1984.

43. Nof. Shimon Y., and Lechtman, Hannan, "Robot Time and Motion System Provides Means of Evaluating Alternative Robot Work Method," Industrial Engineering, Vol. 14, No. 4, 1982.

44. Nof, Shimon Y., Kight, James L., and Salvendy, Grariel S., "Effective Utilization of Industrial Robots-A Job and Skills Analysis Approach," AIIE Transactions, Vol. 12, No. 3, 1980.

45. Offodile, Onyebuchi F., "Design and Analysis of A Coding and Classification System for A Systematic Interactive Computer Aided Robot Selection Procedure

(CARSP)," Unpublished PhD. Dissertation in Industrial Engineering, Texas Tech University, Lubbock, Texas, 1984.

46. Owen, Tony, Assembly with Robots, Prentice-Hall Inc., Englewood, N. J., 1985.

47. Padberg, M. W. and Hong, S., "On the Symmetric Traveling Salesman Problem: A Computational Study," Mathematical Programming Study, Vol. 12, pp. 78-107, 1980.

48. Parker R. G. and Rardin R. L. "The Traveling Salesman Problem: An Update of Research" Naval Research Logistics Quarterly, Vol. 30, pp. 69-96, 1983.

49. Paul, Richard P., and Nof, Shimon Y., "Work Methods Measurement-A Comparison Between Robot and Human Task Performance," Int. J. of Prod. Res., Vol. 17, No. 3, 1979.

50. Randhawa, S. U., McDowell, E. D., and Faruqui, Salahud-Din, "An Integer Programming Application to Solve Sequencer Mix Problems in Printed Circuit Board Production," Int. J. of Prod. Res., Vol. 23, No. 3, 1985.

51. Ranky, P. G., and Ho, C. Y., Robot Modeling, IFS Ltd., England, 1985.

52. Richardson, M. K., "Computer Program Controls Component Sequencer Costs," Assembly Engineering, Vol. 19, No. 12, 1976.

53. Sanderson, Ronald J., et al., Industrial Robots: A Survey and Forecasting for Manufacturing Managers, Tech Tran Corporation, Ill. 1982.

54. Sarin, S. C., and Wilhelm, W. E., "Prototype Models for Two-Dimensional Layout Design of Robot Systems," AIIE Transactions, Vol. 16, No. 3, 1984.

55. Scarlett, J. A., Printed Circuit Boards for Microelectronics, The Marconi Company Ltd., London, 1970.

56. Shapiro, J. F., Algorithms for the Solution of the Optimal Cost and Bottleneck Traveling Salesman Problem, Sc.D. Thesis, Washington University, St. Louis, Missouri, 1966.

57. Stauffer, Robert N., "Robotic Assembly," Robotics Today, Vol. 6, No. 5, Oct. 1984.

58. Titlebaum, Ronald P., "An Insertion Machine for Everyone, Almost," Assembly Engineering, Vol. 19, No. 8, 1976.

59. Warnecke, H., Schraft, R. D., and Schmidt-Streier, U., "Computer-aided Planning of Industrial Robot Application in Workpiece Handling," Proceedings of 1981 International Conference, pp. 349-359. 1983.

60. Warnecke, H., Schraft, R. D., and Schmidt-Streier, U., "Application of Industrial Robots for Assembly -Operations in the Automotive Industry," Proceedings of the 13th International Symposium on Industrial Robots, 1983.

61. Wilhelm, W. E., and Sarin, S. C., " A Structure for Sequencing Robot Activities in Machine Tending Applications," Int. J. of Prod. Res., Vol. 23, No. 1, 1985.

62. Young, R., Yuan, S., and Sun, Y., "Development of a Robot-Centered Workstation," 1982, Fall Industrial Engineering Conference Proceedings.

63. Zollo, S., "SMT Mounts an Offensive," Electronics Week, March 11, 1985.

APPENDIX A

THE MODIFIED EASTMAN ALGORITHM

The logical procedure of the medified Eastman algorithm is summarized as follows:

Step 1: Let CLUB represent current upper bound on the optimal solution of the traveling salesman problem. Let KN = 0.

Step 2: Solve the associated assignment problem by the Hungarian algorithm. The solution provides a lower bound on the optimal solution of the traveling salesman problem. If at least one subtour exists in the solution, go to Step 3; otherwise, the optimal solution of the assignment problem is also optimal of the traveling salesman problem; stop.

Step 3: Select the subtour with the least number of links and let K be the number of links in the selected subtour.

Step 4: If KN = 0 and K = 2, go to Step 5. If KN > 0 and K = 2 go to Step 7. If K > 2, go to Step 8.

Step 5: The subtour is with 2 links. Designate the subtour as I1-I2-I1, then let D(I1, I2) = .

Step 6: Solve the new assignment problem by the Hungarian algorithm. The solution distance is the lower bound

for this subproblem. Set KN = KN + 1 and then go to Step 10.

Step 7: Branch into K subproblems. The subtour is with 2 links. Suppose the subtour is I1-I2-I1, then for subproblem 1 let D(I1, I2) = ∞. Let KN = KN + 1. Go to Step 9.

Step 8: Branch into K subproblems. If the subtour is I1-I2-I3----IK-I1, then for subproblem 1, let D(I1, I2) and D(I2, I1) be ∞ ; for subproblem 2, let D(I2, I3) and D(I3, I2) be ∞ ; ....etc., and for subproblem K, let D(IK, I1) and D(I1, IK) be ∞.

Step 9: Solve the K new assignment problems by the Hugarian method. Each solution distance is the lower bound for the corresponding subproblem.

Step 10: If there are one or more feasible tours for Step 6 or Step 9 and if the smallest total distance for these feasible tours, say STD, is smaller than CLUB, let CLUB = STD and save the corresponding feasible tour. Otherwise CLUB remains unchanged.

Step 11: If CLUB is less than the lower bounds on all the other unexplored subproblems, then the solution corresponding to CLUB is an optimal solution of the traveling salesman problem, so stop; otherwise, go to next step. Unexplored subproblems are those

subproblems that have not been divided into further subproblems.

Step 12: From the set of all unexplored nonfeasible subproblems (subtours present), with a lower bound less than CLUB, select the subproblem with the smallest lower bound for further branching. Go to Step 3.

APPENDIX B

DATA OF THE 109-COMPONENT BOARD

COMPONENT IDENTIFICATION NUMBER	X COORDINATE (In.)	Y COORDINATE (In.)	COMPONENT TYPE IDENTIFICATION NUMBER
1	1.2	1.8	1
2	1.3	2.9	2
3	1.3	2.4	3
4	1.2	6.8	4
5	4.6	2.2	5
6	4.2	3.4	6
7	4.2	3.9	7
8	0.7	6.8	8
9	4.4	5.5	9
10	5.6	9.8	10
11	0.7	1.9	11
12	2.8	9.9	12
13	3.9	5.5	13
14	4.2	7.1	14
15	3.8	7.1	15
16	3.3	0.2	16
17	5.0	8.8	17
18	4.9	5.0	18
19	2.4	2.4	19
20	0.2	4.6	20
21	3.7	3.4	21
22	3.4	8.2	22
23	0.7	1.4	23
24	5.0	7.1	24
25	1.3	9.2	25
26	1.3	9.6	26
27	1.4	7.4	27
28	2.9	2.1	28
29	2.3	8.7	29
30	1.9	2.3	30
31	3.9	9.3	31
32	2.8	8.7	32
33	3.4	9.3	33
34	1.3	8.6	34
35	3.4	8.7	35
36	4.0	8.1	36
37	2.9	7.4	37

38	2.9	2.7	38
39	1.7	1.7	39
40	4.0	8.7	40
41	4.4	4.9	41
42	0.2	2.3	42
43	1.9	7.4	43
44	4.5	9.8	44
45	3.9	2.1	45
46	5.2	2.7	46
47	1.7	6.9	47
48	4.5	8.8	13
49	3.7	6.5	14
50	2.3	2.9	15
51	1.7	2.9	15
52	2.7	0.2	16
53	3.9	0.3	16
54	3.6	4.4	17
55	3.7	3.9	17
56	3.9	5.0	17
57	4.5	8.2	17
58	5.0	8.2	17
59	2.4	0.4	18
60	2.2	1.8	19
61	2.4	7.5	19
62	0.2	6.9	20
63	0.2	4.0	20
64	0.2	2.9	20
65	0.2	6.3	20
66	0.2	0.8	20
67	0.2	3.4	20
68	0.2	5.1	20
69	0.2	8.8	20
70	4.7	0.4	20
71	0.2	1.9	20
72	0.2	5.7	20
73	1.3	0.4	20
74	0.2	1.3	20
75	1.9	8.2	20
76	0.8	2.9	21
77	0.8	2.4	21
78	0.7	0.9	23
79	0.7	0.4	23
80	0.2	9.3	25
81	1.4	8.1	27
82	3.4	2.1	28
83	4.9	5.4	28
84	4.3	6.5	28
85	3.4	2.7	28
86	2.9	8.1	28

87	2.4	8.1	29
88	2.9	6.9	29
89	0.7	8.7	30
90	3.9	9.8	30
91	3.4	6.9	35
92	5.1	9.8	36
93	1.8	8.7	36
94	3.4	9.8	36
95	1.9	0.3	36
96	5.0	7.6	36
97	3.4	7.4	42
98	2.8	9.3	42
99	3.9	2.7	45
100	4.5	2.7	46
101	5.5	8.4	46
102	5.2	0.9	46
103	5.5	9.0	46
104	5.5	7.1	46
105	5.2	2.1	46
106	5.2	1.5	46
107	5.5	7.8	46
108	4.5	9.3	46
109	0.7	9.2	31